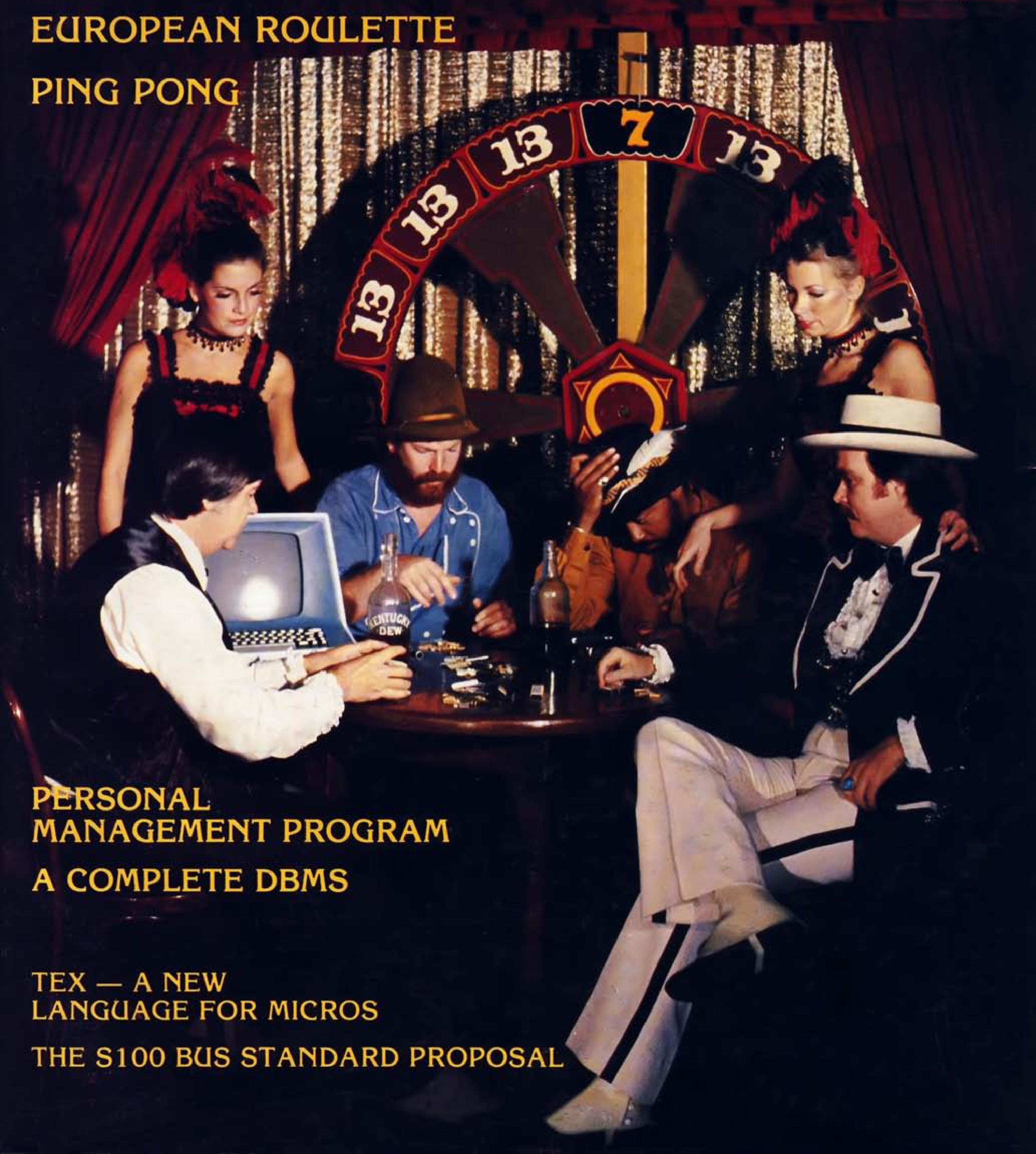
INTERFACE AGE™

COMPUTING FOR HOME AND BUSINESS APPLICATIONS

VOLUME 3, ISSUE 8 AUGUST 1978 $2.00
CANADA/MEXICO $2.50 INTERNATIONAL $3.50

EUROPEAN ROULETTE

PING PONG

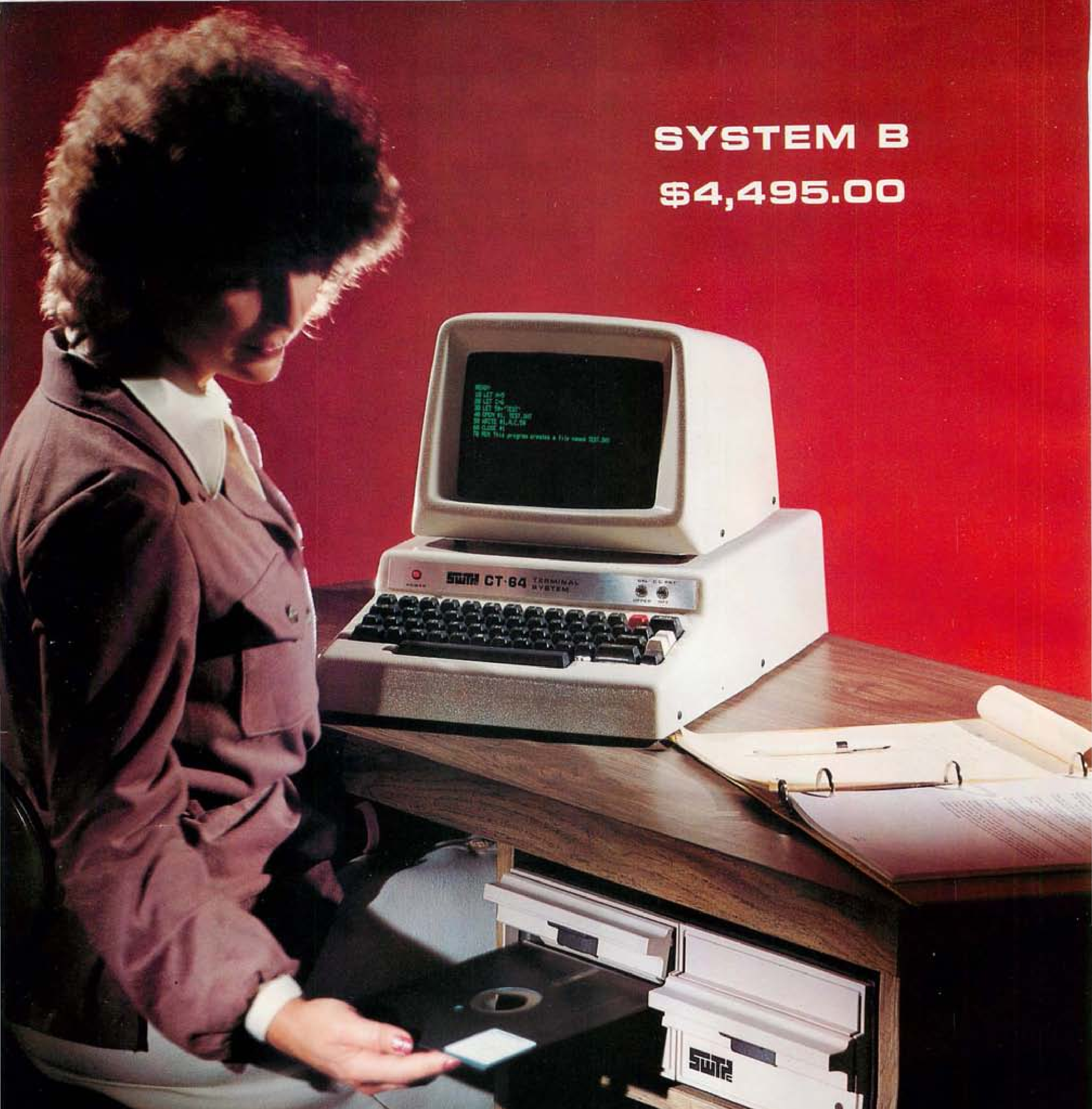


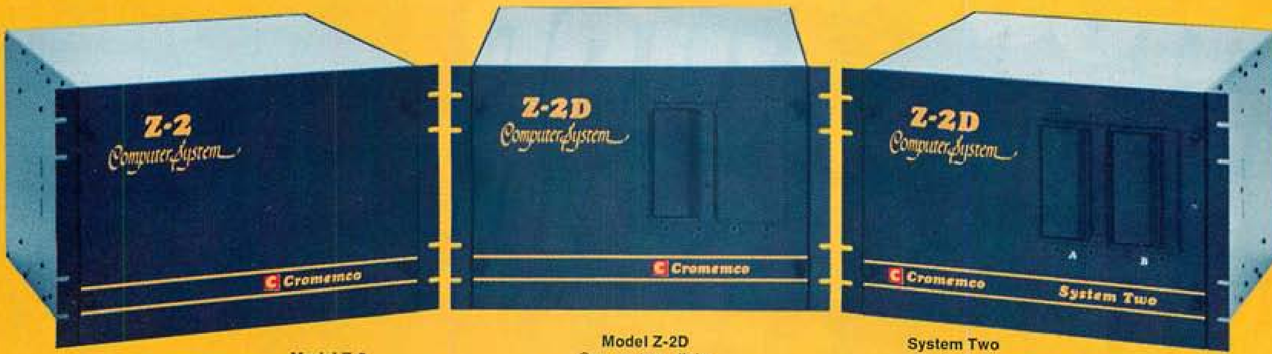
PERSONAL
MANAGEMENT PROGRAM

A COMPLETE DBMS

TEX — A NEW
LANGUAGE FOR MICROS

THE S100 BUS STANDARD PROPOSAL


SYSTEM B
$4,495.00
CT-64



Model Z-2
Up to 512K of RAM/ROM

Model Z-2D
One or two disks
Up to 512K of RAM/ROM
Up to 184K of disk

System Two
Dual disk
Up to 512K of RAM/ROM
Up to 184K of disk

Fill your computer needs with the industry's most professional microcomputers

#1 IN RELIABILITY

When you choose Cromemco you get not only the industry's finest microcomputers but also the industry's widest microcomputer selection.

What's more, you get a computer from the manufacturer that computer dealers rate #1 in product reliability.*

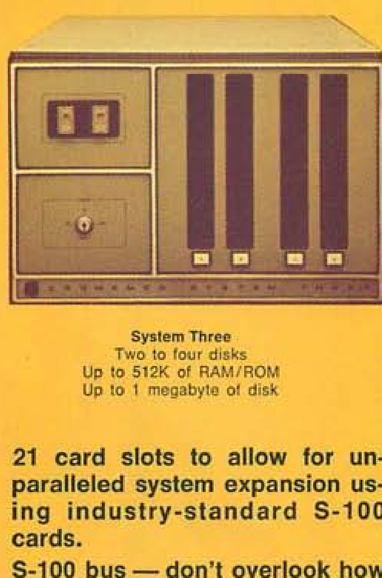
Your range of choice includes our advanced System Three with up to four 8″ disk drives. Or choose from the System Two and Z-2D with 5″ drives. Then for ROM-based work there's the Z2. Each of these computers further offers up to ½ megabyte of RAM (or ROM).

We say these are the industry's most professional microcomputers because they have outstanding features like these:

- **Z-80A microprocessor — operates at 250 nano second cycle time — nearly twice the speed of most others.**

*Rated in ***The 1977 Computer Store Survey*** by Image Resources, Westlake Village, CA.

Up to 512 kilobytes of RAM and 1 megabyte of disk storage



System Three
Two to four disks
Up to 512K of RAM/ROM
Up to 1 megabyte of disk

- **21 card slots to allow for unparalleled system expansion using industry-standard S-100 cards.**
- **S-100 bus — don't overlook how important this is. It has the industry's widest support and Cromemco has professionally implemented it in a fully-shielded design.**
- **Cromemco card support of more than a dozen circuit cards for process control, business systems, and data acquisition including cards for A-D and D-A conversion, for interfacing daisy-wheel or dot-matrix printers, even a card for programming PROMs.**
- **The industry's most professional software support, including FORTRAN IV, 16K Disk-Extended BASIC, Z-80 Macro Assembler, Cromemco Multi-User Operating System — and more coming.**
- **Rugged, professional all-metal construction for rack (or bench or floor cabinet) mounting. Cabinets available.**

FOR TODAY AND TOMORROW

Cromemco computers will meet your needs now and in the future because of their unquestioned technical leadership, professionalism and enormous expandability.

See them today at your dealer. There's no substitute for getting the best.

Cromemco
incorporated
Specialists in computers and peripherals
280 BERNARDO AVE., MOUNTAIN VIEW, CA 94040 • (415) 964-7400

CIRCLE INQUIRY NO. 12

VOL. 3, ISSUE 8
SINCE DECEMBER 1975

AUGUST 1978

INTERFACE AGE™

COMPUTING FOR HOME AND BUSINESS APPLICATIONS

THIS MONTH'S COVER

This month's cover represents what might have been had the saloons of the Wild West had the advantage of microcomputers. Every table would have had a microdealer, and if gamblers had a good night, they went home with saddle bags loaded with integrated chips.

The cover was designed by our Art Director Fino Ortiz; photography by Shelley Wright. We would like to thank Knott's Berry Farm of Buena Park, California for their cooperation and for the use of their Calico Saloon and personnel. The terminal was courtesy of Lear Siegler and the integrated chips were supplied by Rockwell.

GENERAL FEATURES

BUSINESS FEATURES

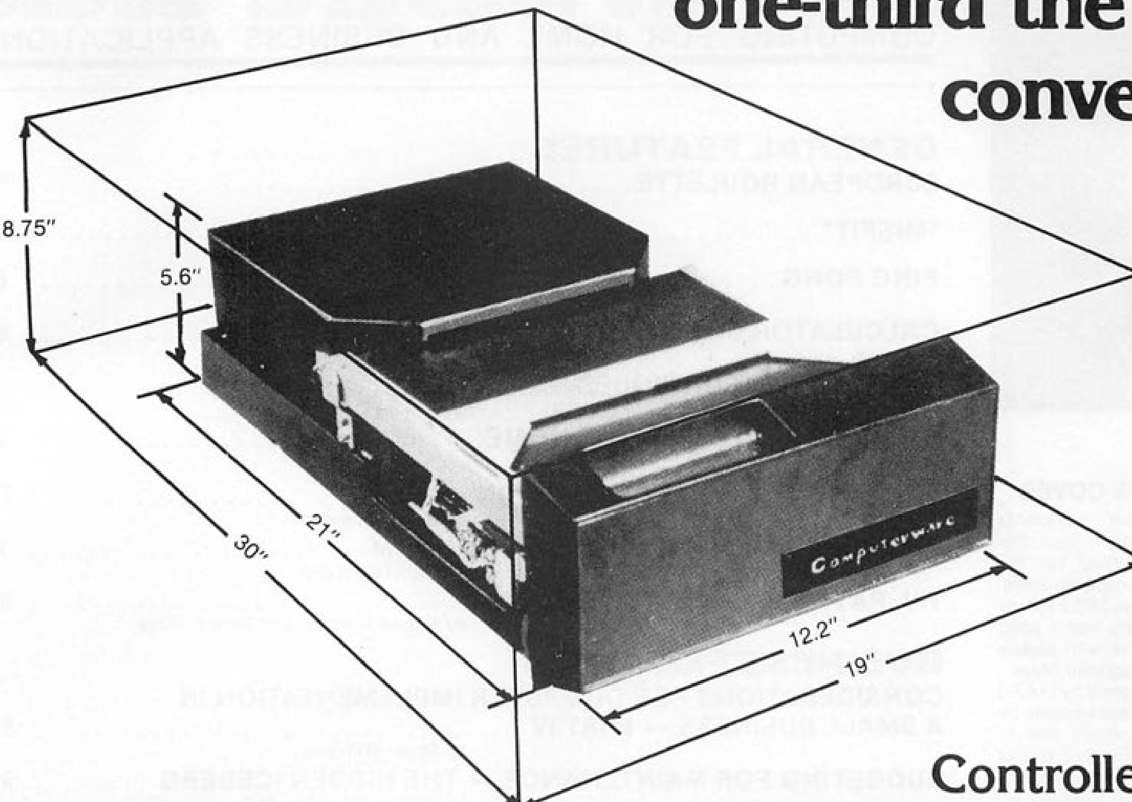
HARDWARE FEATURES

SOFTWARE FEATURES

INTERFACE AGE Magazine, published monthly by McPheters, Wolfe & Jones, 16704 Marquardt Ave., Cerritos, CA 90701. Subscription rates: U.S. $14.00, Canada/Mexico $16.00, all other countries $24.00. Make checks payable in U.S. funds drawn on a U.S. bank. Opinions expressed in by-lined articles do not necessarily reflect the opinion of this magazine or the publisher. Mention of products by trade name in editorial material or advertisements contained herein in no way constitutes endorsement of the product or products by this magazine or the publisher.

INTERFACE AGE Magazine is catalogued in the Library of Congress, Classification No. QA75.5.155. Membership in Audit Bureau of Circulations applied for.
POSTMASTER: Please send change of address form 3579 and undelivered copies to **INTERFACE AGE** Magazine, 16704 Marquardt Ave., Cerritos, CA 90701. Second-class postage paid at Artesia, California 90701 and at additional mailing offices.

New disk drive packs 10 Megabytes in unit one-third the size of conventional drives.



Controller Card and S100 Bus Interface Available

Controller for Microcomputer Systems

Among the many applications of the D120, D140 mididisk drives one in particular is the disk drive used in association with a microprocessor-based microcomputer system.
The addition of a controller to a disk drive constitutes a disk subsystem which is easily connectible to a microprocessor-based user system using an 8 bit input/output architecture bus.
The ease of an interface connection is readily evident from various tasks managed by the controller, namely:

— address management
— sequential control of read/write operations
— asynchronous data transfer by user in buffered mode
— synchronous data transfer in real time
— buffered data memory with 4 sector capacity
— autonomous processing of defective sectors on media
— format processing

Main Features

The controller is designed to work in three different modes of operation—buffered—real time, internal halt—and real time, external halt.

Buffered Mode

From one to four consecutive sectros are stored in a memory buffer. Memory capacity is IK—i.e., 4 x 256 bytes. Data transfer is made in asynchronous mode. If, during a write operation, the drive signals a media defect on one of the addressed sectors, the controller ensures correct defective sector processing by using the spare sectors reserved for this purpose at the end of the track.

Real Time - internal halt

This second mode of operation requires no buffer. The user-system initiates a read or write operation for one to four consecutive sectors. The controller manages the entire operation and stops automatically as soon as the requested number of sectors has been processed.
Data transfer is synchronous, at a rate of 916 KBytes/Second with an 8 bit wide data path, (—parity bit) carried out in real time.
Defective sector management can be carried out in two ways. Firstly, if a defective sector is detected, data is written into the sector following.
Management is ensured by the controller. Alternatively, defective sectors may be handled by the user. In this case the controller stops at the faulty sector, dispatches a status flag over the interface to the user-system which can then process the defective sector by software; i.e., go to spare sector at the end of the track, or use another spare track reserved for this purpose, (generally, track 000).

Real Time - external halt

This mode of operation is also executed without the buffer memory. A read/write operation is initialized by the user-system for any number of sectors on the same track address. External halt is effected when the system signals to the controller an end of operation. In this way, it is possible to write one complete track, or, read a track several times. Data transfer and defective sector management are carried out in exactly the same way as in Real Time, internal halt—see above.
Note: The different modes of operation described above are selected during manufacture. One controller is required per disk drive.

EDITOR'S NOTEBOOK

The last several weeks have been busy and exciting. Also during this period some events have taken place that have made being an editor more than a little difficult. Part of this is due to some of the letters we have been receiving here at the Cerritos office.

First, INTERFACE AGE is a magazine — we report on the industry. We don't manufacture anything, not even the news. We have had a number of letters asking us for information on our interfaces to printers, etc. Please, readers, look at our new products section. You will be surprised at the amount of information you can find.

Next, we have a misunderstanding with some clubs regarding our calendar section. The calendar is a service we offer to let people know what clubs meet where and when. Unfortunately we have published some wrong dates. We publish the dates that are sent to us, and we have set up a policy that we have to have the date sent approximately 60 days in advance of a given issue. For example, in order to publish the correct date of a club meeting in the October issue, we need the information by the first week of August.

You have probably noticed that we don't publish the addresses of authors in the magazine. We do this basically to protect the author — many have expressed a fear of losing their equipment. Now this may create a problem for those of you who wish to contact the author for some reason or another. To do so send us the letter with a stamped envelope and we will forward it directly to the author, **but please do not call us.**

ERRORS

Occasionally an error will pop up in an article, and we get a number of letters and phone calls about it. This was the case with the 24-line display in the June issue. Mr. Sama quickly noted our error and has sent us the correction which you will find in the letters to the editor section.

We do try to do all possible to avoid printing errors, especially ones that are totally incorrect as stated in an article. We therefore employ a number of design engineers and software consultants to assist in the evaluation of articles.

In the case of the Sama article the design was thoroughly checked before publication — the error was ours. We just recently finished our Heath terminal and did the modification. The only problem we encountered was due to a cold solder joint. Otherwise we have had no other problems.

HAPPENING

This year's National Computer Conference, held June 5-8 at the Anaheim Convention Center and Disneyland Hotel, was better than many thought it would or could be. Over 59,000 people attended and the old excitement seemed to be there.

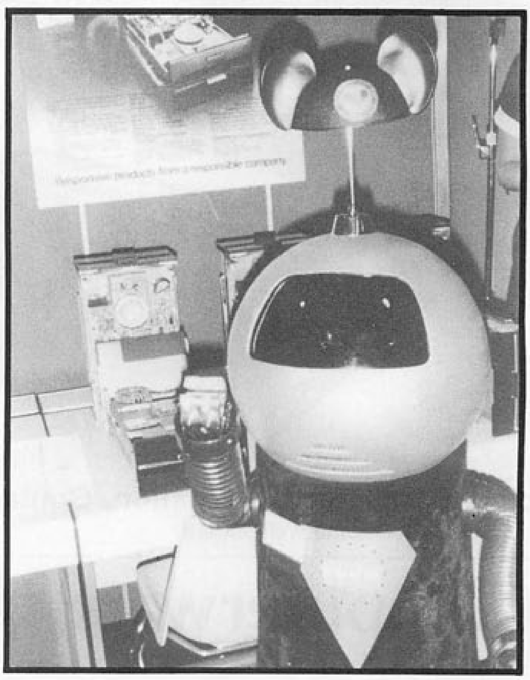
The main show was in the convention center in four display areas housing over 1500 booths. Almost every manufacturer of hardware and software was exhibiting and providing some very exciting inputs to what is happening in both the mainframe and micro industry.

INTERFACE AGE HAS CLOSE ENCOUNTER

Last month you probably saw the short article we had on ORION, with the promise to tell more about it. ORION is an animated robot that carries on a conversation with the public.

While at the NCC I had a chance to meet the metal comedian and was truly impressed. ORION is the brainchild of Ron and Terry Palmer who market his — or is it her — talents to companies who wish to draw crowds at conventions. And draw crowds it does.

The owners did divulge the secret of ORION to INTERFACE AGE in an exclusive interview but asked that it not be given away. According to Terry Palmer, Vice President of Digi Tech, giving away the secret is like shooting Santa Claus. "What we are trying to do," she went on, "is to provide a little fun in life. We make no false claims about ORION, but feel we are a highly sophisticated magic act."



Look To The North Star HORIZON Computer.

HORIZON™— a complete, high-performance microprocessor system with integrated floppy disk memory. HORIZON is attractive, professionally engineered, and ideal for business, educational and personal applications.

To begin programming in extended BASIC, merely add a CRT or hard-copy terminal. HORIZON-1 includes a Z80A processor, 16K RAM, minifloppy™ disk and 12-slot S-100 motherboard with serial terminal interface — all standard equipment.

WHAT ABOUT PERFORMANCE?

The Z80A processor operates at 4MHZ — double the power of the 8080. And our 16K RAM board lets the Z80A *execute at full speed.* HORIZON can load or save a 10K byte disk program in less than 2 seconds. Each diskette can store 90K bytes.

AND SOFTWARE, TOO

HORIZON includes the North Star Disk Operating System and full extended BASIC on diskette ready at power-on. Our BASIC, now in widespread use, has everything desired in a BASIC, including sequential and random disk files, formatted output, a powerful line editor, strings, machine language CALL and more.

EXPAND YOUR HORIZON

Also available—Hardware floating point board (FPB); additional 16K memory boards with parity option. Add a second disk drive and you have HORIZON-2. Economical serial and parallel I/O ports may be installed on the motherboard. Many widely available S-100 bus peripheral boards can be added to HORIZON.

QUALITY AT THE RIGHT PRICE

HORIZON processor board, RAM, FPB and MICRO DISK SYSTEM can be bought separately for either Z80 or 8080 S-100 bus systems.

HORIZON-1 $1599 kit; $1899 assembled.
HORIZON-2 $1999 kit; $2349 assembled.

16K RAM—$399 kit; $459 assembled; Parity option $39 kit; $59 assembled. FPB $259 kit; $359 assembled. Z80 board $199 kit; $259 assembled. Prices subject to change. HORIZON offered in choice of wood or blue metal cover at no extra charge.

Write for free color catalogue or visit your local computer store.

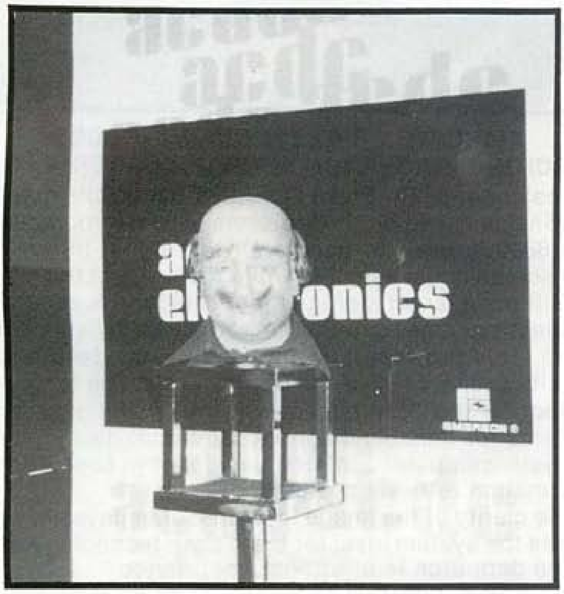
NORTH STAR ★ COMPUTERS
2547 Ninth Street • Berkeley, California 94710 • (415) 549-0858

CIRCLE INQUIRY NO. 38

Yes INTERFACE AGE knows the secret and it will be kept, but no ORION is not microprocessor controlled nor does he possess a large vocabulary. But ORION is fantastic just the same. For those of you who are interested in finding out about using ORION at a convention or sales meeting contact: Terry Palmer, Vice President, Digi Tech, Inc., 58 Van Buren Avenue, Metuchen, NJ 08840, phone (201) 548-4260.

acdc ELECTRONICS HAS HEAD ON STRAIGHT

One of the crowd pleasers that was found at the NCC was Sparky Watts the talking head, used by acdc electronics to tell visitors about the sub-modular switching power supply. Sparky is supported by two technicians who control the voice, mouth movements and facial expressions. An interviewer is part of the show that leads the discussion of the virtues and benefits of the power supply line featured by acdc.



Sparky talks about low-cost open frame linears, high-quality modular OEM linears and high performance fan-cooled switches. acdc provides some of the best designed and cost effective power systems to the computer industry today. Additional information on their product line can be obtained by contacting Bob Hecton, acdc electronics, Division of Emerson Electric Co., 401 Jones Road, Oceanside, CA 92054, phone (714) 757-1880.

DRAGON, THE HONEYWELL THEME

Among the minicomputer exhibitors was Honeywell Information Systems featuring their Level 6 family of minicomputers and terminals. The systems are designed to provide the OEM and system builder with a high degree of flexibility. The system is based on a unique 6 million byte per second megabus. The system is really top of the line for minicomputer systems.

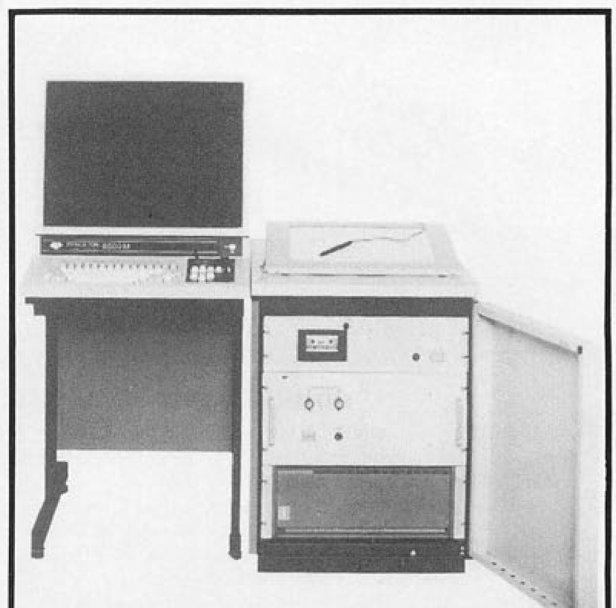
To tell everyone about the system and the problems it solves, the Honeywell team used a smart aleck dragon who had an eye for the girls. The talk was informative and fun to listen to. The Honeywell folks can be contacted by writing to Carol L. Levine, Honeywell Information Systems, 300 Concord Road, Billerica, MA 01821, phone (617) 667-3111.

PRINCETON ELECTRONIC INTRODUCES EXCITING INNOVATION IN GRAPHICS TERMINALS

Designed for Dr. Steve Hofstein, the 8500M graphics terminal is based on the Motorola 6800 microprocessor. The 6800 is used to handle all the digital information and screen housekeeping, while the heart of the system is a lithocon target storage tube. This tube is essentially an electronic memory that is addressed by an electron beam. This provides a 5K by 5K by 8 bit deep display density and up to 32 levels of gray scale. The terminal is perfect for the OEM engaged in the design of graphics systems for medical diagnostics, circuit design, image analysis, computer assisted design, or in sophisticated information retrieval systems.

The clarity of the image is almost unbelievable, which makes the system ideal for brain scan technology where image definition is of utmost importance.

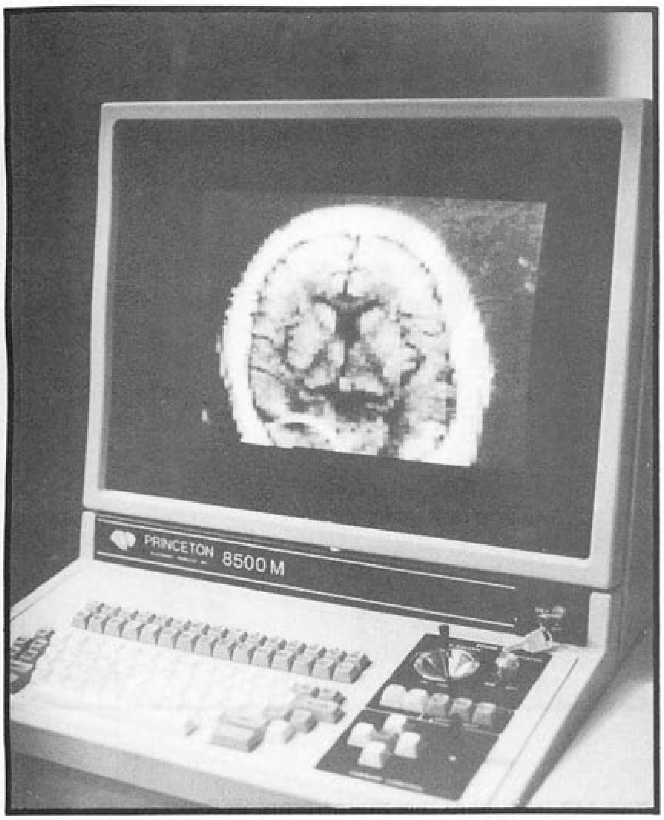
For further information on this exciting terminal contact Don Zigo, Princeton Electronic Products, Inc., P.O. Box 101, North Brunswick, NJ 08902, phone (201) 297-4448



ATTENTION!

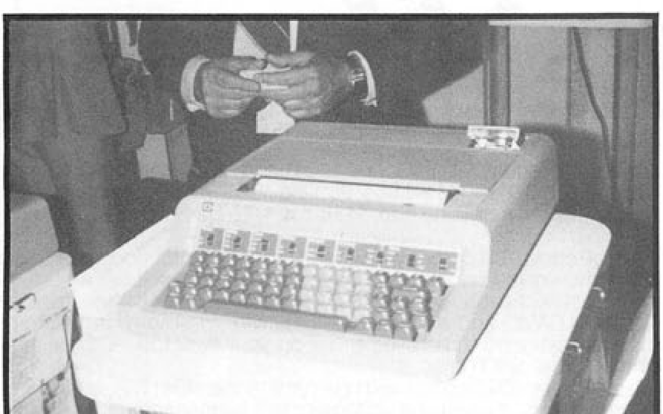
Due to a typographical error in the July **Electronic Control Technology** *ad, the price for the I6K Kit was listed as $239 instead of the correct price of $350. We apologize for any inconvenience this may have caused our readers.*

INTERFACE AGE



COMPUTER DEVICES HAS PORTABLE TERMINAL

Computer Devices of Burlington, Maine, exhibited their complete line of MINITERM terminals and printers. The MINITERM series ASR terminals feature 8K of memory which is expandable to 32K, with built in cassette tape and dual 6800's to handle the necessary functions both on and off line. The terminal makes it possible to do off line editing and prepare information to dump to a larger system at some later time. The terminals have both upper and lower case, full ASCII character set, and sell in the 3500 to 3800 dollar range.



The 1201 RO MINITERM thermal printer is equipped to handle both serial and parallel interfaces. Paper costs approximately $2.00 per roll of 100 feet. Selling in the 1400 dollar range, the printer can be buffered to handle a 2 to 4K buffer with receive burst up to 9600 baud. The printer has a 50 cps maximum printing rate. The printer is quiet, fast and reliable.

For small businessmen looking for the system to interface to a time share service or OEMs who are building complete systems the MINITERM product line should fit the bill. Further information can be obtained by writing to: Donald R. Cadieux, Computer Devices Inc., 25 North Avenue, Burlington, ME 01803, phone (617) 273-1550

VECTOR GRAPHIC

Vector Graphic is a computer company that was started August 20, 1975 by two women; Lori Harp and Carole Ely, who had an idea and a dream.

Beginning with a memory board for the MITS Altair, which was introduced in the pages of INTERFACE AGE, the company has grown from a two-woman operation, in a living room, to a highly respected and dynamic company.

The key to this growth is the marketing know-how and industry insight Harp and Ely have developed. As Lori Harp put it, in a recent interview: "When other companies were promising deliveries we delivered. Our marketing plan called for establishing distributors and providing our product as quickly as possible — which in the early days usually meant about three to four days." Now with the rapid expansion of the company, delivery is still of key importance.

Recently Vector Graphic introduced a new system called the Vector MZ, which is a total system selling for $3,750. This system comes fully assembled and burnt in.

The MZ includes the mainframe, Z80 processor board running at 4MHz, I/O board for both serial and parallel operation, 2708 PROM board with one monitor PROM, disk controller board, and two 630K Micropolis disk drives. Also available is extended disk BASIC with a DOS. CP/M is also available for an additional $100.

Vector feels the M2 is the ideal business system and is using the third party concept in application development. There is very sound business reasoning behind this decision. Vector Graphic is a hardware manufacturer and at this point in time their main concern is in providing the best possible hardware. Both Lori and Carole feel that should they go into the software development business at this point in time the company's efforts would be diluted.

However, they do have plans of eventually tackling the software market. Probably in the area of application software.

For those of you interested in finding out more about the M2 system or other Vector Graphic products, contact: Carole Ely, Vector Graphic, Inc., 790 Hampshire Road, Westlake Village, CA 91361, phone (805) 497-6853

SOFTWARE IS THE KEY TO THE INDUSTRY

Traditionally the computer industry has suffered from the lack of software in all areas. Over the years the mainframers and mini makers have overcome some of the major problems in systems software, languages and applications. One company, Dataware, has developed an interesting approach to mainframe software and that is of conversion. They have worked out methods of converting from RPG/RPG II to PL/1, PL/1 to COBOL and other translator packages. The important point of mentioning this company here is the concept they have developed.

Although Dataware provides services to the large system folks they are interesting to talk to and their brochures are helpful in understanding translation concepts. They can be reached by contacting: Robert Dinkel, Dataware Inc., 495 Delaware St., Tonawanda, NY 14150, phone (716) 695-1412

INTERFACE AGE

16704 Marquardt Avenue, Cerritos, CA 90701
(213) 926-9544

PUBLISHER & EDITOR-IN-CHIEF **ROBERT S. JONES**
EXECUTIVE PUBLISHER **NANCY A. JONES**
ASSISTANT-TO-THE-PUBLISHER **KATHY ROBERTSON**
GENERAL MANAGER **EVA YAKA**
MARKETING MANAGER **MEL MURCHISON**
MARKETING ASSISTANT **DENISE JACKSON**

ADMINISTRATION
PUBLICATION DIRECTOR **MIKE ANTICH**
CIRCULATION MANAGER **DONNA YOSHIOKA**
SUBSCRIPTION CIRCULATION **JO ANN FERGUSON**
DOMESTIC RETAIL CIRCULATION **ZACH BOVINETTE**
(213) 795-7002
ACCOUNTING ASSISTANT **SAYOKO TANISAKI**

EDITORIAL
SENIOR EDITOR **CARL WARREN**
ASSISTANT EDITOR **SANDI EVANS**
NORTHWESTERN REGIONAL EDITOR **ADAM OSBORNE, PhD**
NORTHEASTERN REGIONAL EDITOR **ROGER C. GARRETT**
SOUTHEASTERN REGIONAL EDITOR **BILL TURNER**
HARDWARE EDITOR **ROGER EDELSON**
CONTRIBUTING BUSINESS EDITOR **BUD SHAMBURGER**
CONTRIBUTING EDITOR **ALAN MILLER**

Editorial Correspondence
Direct all correspondence to the appropriate editor at: INTERFACE AGE Magazine, P.O. Box 1234, Cerritos, CA 90701.

PRODUCTION
PRODUCTION MANAGER **MARGARET FENSTERMAKER**
ASSISTANT PRODUCTION MANAGER **SHELLEY WRIGHT**
ART DIRECTOR **FINO ORTIZ**
TYPOGRAPHER **MELODY A. MARTENS**

ADVERTISING
NEW ENGLAND REGION **DICK GREEN**
7 Lincoln St., Wakefield, MA 01880 (617) 246-1986

EASTERN REGION **TONY CARLSON**
20 Community Pl., Morristown, NJ 07960 (201) 267-3032

MIDWEST REGION **AL GRAVENHORST & STEVE SKINNER**
5901 N. Cicero Ave., Chicago, IL 60046 (312) 545-8621

WESTERN REGION **BRUCE BERKEY & ZACH BOVINETTE**
61 S. Lake Ave., Pasadena, CA 91106 (213) 795-7002

COMPUTER RETAIL STORES NATIONWIDE
CALL (213) 795-7002 (COLLECT)

FOREIGN CIRCULATION
JAPAN CIRCULATION **KAZUHIKO NISHI**
ASCII Publishing - 305 HI TORIO, 5-6-4 Minami Aoyama, Minato-ku, Tokyo 107 Japan
Telephone: (03) 407-4910

UNITED KINGDOM CIRCULATION **VINCENT COEN**
L.P. Enterprise, 313 Kingston Road, Ilford, Essex, England IGIIPJ.
Telephone: 01-553-1001

WESTERN CANADA CIRCULATION **BRIAN I.J. WIEBE**
Kitronic, 25236 26th Avenue RR5, Aldergrove, B.C. VOX 1A0
Telephone: (604) 856-2301

EASTERN CANADA CIRCULATION **LIZ JANEK**
RS-232, Ltd., 186 Queen Street West, Toronto, Ontario M5V 1Z1
Telephone: (426) 598-0288

AUSTRALIA CIRCULATION **R. J. HOESS**
Electronic Concepts Pty. Ltd., 52-58 Clarence Street, Sydney NSW 2000
Telephone: 29-3753

INDONESIA, SINGAPORE, MALAYSIA CIRCULATION **LEE MILES**
The Computer Centre, Pte. Ltd., 5366, Woh Hup Complex, Beach Rd.
Singapore 7 293-2630

MEMBER OF THE WESTERN PUBLISHERS ASSOCIATION

AMERICAN SOCIETY OF BUSINESS PRESS EDITORS

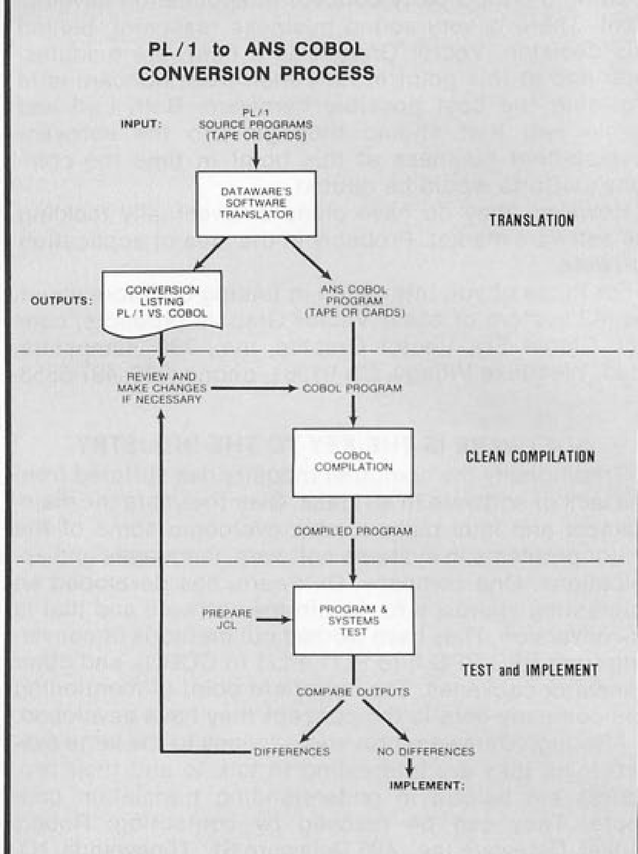


Figure 1

BASIC/FOUR AIMS FOR MEDICAL WORLD

Basic/Four Corporation, one of the leaders in business systems, has developed a unique hardware/software package aimed at the medical world. MEDIFACS™ is a financial and statistical data processing system for small and medium hospitals. The system is especially designed to handle the wide ranging functions found in hospitals from admissions to financial reporting.

MEDIFACS is designed with the future in mind and human engineered. Every aspect of medical accounting and paperwork has been taken into account, probably the best medical system on the market today. Contact: Judi Williams, Basic/Four Corp., P.O. Box C-11921, Santa Ana, CA 92711, (714) 731-5100.

ANOTHER CONSULTANT IS ON THE SCENE

Last month I talked about software consultants and predicted that this is the growing field to watch. Well, I must be a little bit correct. At the NCC I had the chance to meet two extremely adept fellows who gear themselves toward the Alpha Micro systems. They are Henry Cordova and Duane Cowgill. After talking to them for quite a long time I got the impression that they have an excellent future in store. For those of you who wish to contact these gentlemen contact ON-TRAK Software Consultants, 1009 E. Mt. Curve St., Altadena, CA 91001, phone (213) 794-1439.

Speaking of making requests. Are you in the consultant business? If so, let me hear from you. I want to know who you are, where you are, and where you plan to go. Send your letters to Carl Warren, INTERFACE AGE Magazine, P.O. Box 1234, Cerritos, CA 90701. **Please write — no phone calls.**

—carl

LETTERS TO THE EDITOR

Dear Editor:

The stuff Adam Osborne writes usually indicates clear thinking. But his article in the May issue is indicative of thought gone awry. I find it unacceptable that he would defend editorial silence concerning Associated Electronics' malfunctioning product line and their subsequent bankruptcy. It is evident that Osborne didn't lose money in the fiasco. I did and I'm bitter. Getting the products C.O.D. did me no good. (Reference Osborne's [valid] point about the impropriety of providing interest-free loans to firms who accept your payment and deliver months later.)

The PROM programmer system I purchased from Associated never did function properly.

I'm sure others, who like myself lost half a grand or so, *do not share Osborne's enthusiasm for Bryce Ward's future success.* Osborne must learn not to wander out of his field of expertise when he writes — he will quickly lose the confidence of his readers and they will desert him.

Will you kindly publish the name of the new business Bryce Ward has just founded?

Jack M. Williams

DR. OSBORNE ANSWERS

Dear Editor:

Mr. Jack Williams has raised a very crucial point in his letter. In my opinion Mr. Williams' letter is one of the most important that I have seen in any of the personal computing magazines. I really appreciate Mr. Williams taking the time to write it and INTERFACE AGE Magazine giving it the publication space.

However, I will not change my position. My purpose in responding here is to defend my position.

Bryce Ward got into trouble at Associated Electronics by underestimating the time it would take to design and debug various boards. Associated Electronics got into trouble because Bryce Ward, in all good faith, accepted money for products he thought he would be able to ship in a few weeks — then discovered he could not. Bryce Ward almost broke his health trying to make good on his commitments; he finally gave up only when physically and financially forced to do so.

At no point, so far as I have been able to determine, did Bryce Ward do anything knowingly dishonest. But more important, at no point did he do anything which was unusual among manufacturers. Some were luckier than others and that is all. If I published the names of all companies who had, and still do operate in the fashion of Associated Electronics, I would have to publish the names of more than half the manufacturers active today. Would Mr. Williams really like me to put all of these companies out of business? A majority of them are likely to survive; surely they should be given this survival chance. Only a minority will fold, as did Associated Electronics, because the market is still so heavily a seller's market.

I maintain that I do the most people the most good when I name only those companies who are knowingly and intentionally committing fraud. I most certainly will name any such companies, providing I have suitable evidence. Meanwhile I will continue my campaign to warn customers that more than half the manufacturers building microcomputer hardware today are in a fiscally precarious situation, not far removed from that of Associated Electronics. Moreover, these manufacturers will continue to act irresponsibly as long as people such as Mr. Williams continue to pay cash in advance, or even C.O.D., when buying from unknown sources. In my opinion the Jack Williams's of this industry are just as irresponsible, and just as much to blame, as are the manufacturers who indulge in forward financing.

It is also worth noting that this whole industry started with inexperienced amateurs doing a job which none of the established manufacturers saw fit to do. Do we not owe these enterprising individuals something? Had we left it to DEC and Data General, there would be no computer stores today and no $300 CPU boards. When these entrepreneurs of the microcomputer industry began, they had no alternative but to operate in a fiscally unsound fashion, since venture capital was unavailable. I have mounted my campaign for fiscal responsibility among manufacturers because venture capital is now available. A year from now, if there are few "hold out" manufacturers who continue to offer disreputable products because they simply choose to adopt unsound fiscal practices, then perhaps I will change my position and I will start naming names.

What I am saying, Mr. Williams, is that a year from now perhaps naming names would be more productive than counterproductive; but today it would be extremely damaging to the whole industry.

Adam Osborne
Osborne & Associates

Dear Editor:

Although INTERFACE AGE usually comes up with sage advice for the computer purchaser, I feel that I must take exception to Adam Osborne's comments in the April 1978 issue concerning the customer-manufacturer interface.

My main objection is your recommendation that everything be purchased through computer stores. This results in increased costs to the customer. We have often received comments from stores that our prices were not high enough for our unique products. In fact, one representative of a major store chain mentioned to me at the West Coast Computer Faire that the price of our programmable character generator was at least $50 under what it should be for stores to carry the product. And this on a product with a 100% success rate to date.

A second objection is your advice not to prepay. Although we accept credit cards and C.O.D. orders, we do ship prepaid product orders first on the basis that those customers trust us. The other problem is the high percentage, 42% for our firm, that are returned when shipped C.O.D., thus increasing the costs for everyone.

We've gathered the family to show you why PERCOM's™ Number 1 in cassette data systems for microcomputers.

Pardon us for doing a little boasting, but we're proud of our family. Proud of each member's reputation for performance and reliability. And pleased that we can offer the best in cassette data systems and data terminal interfacing at low, home-computing prices.

It took more than guts and a little luck to forge a position of leadership. We're number 1 because you get more when you buy PERCOM™. The reason, simply, is experience. Every product described in this ad is based on nearly 10 years of crucial involvement in the design and manufacture of computer peripherals that use cassettes for mass storage.

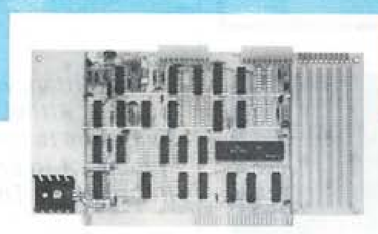
Experience. It's why we developed a more reliable data cassette for home computing. Why our interfacing units provide **both** cassette and data terminal interfacing. Why you get the fastest, most reliable cassette data rates from PERCOM™. Experience. It's *the* reason for PERCOM™.

For your SS-50 bus computer — the CIS-30+

- Interface to data terminal and *two* cassette recorders with a unit only 1/10 the size of SWTP's AC-30.
- Select 30, 60, or 120 bytes per second cassette interfacing, 300, 600 or 1200 baud data terminal interfacing.
- Optional mod kits make CIS-30+ work with *any* microcomputer. (For MITS 680b, ask for Tech Memo TM-CIS-30+—09.)
- KC-Standard/Bi-Phase-M (double frequency) cassette data encoding. Dependable self-clocking operation.
- Ordinary functions may be accomplished with 6800 Mikbug™ monitor.
- Prices: Kit, $79.95; Assembled, $99.95.

Prices include a comprehensive instruction manual. Also available: Test Cassette, Remote Control Kit (for program control of recorders), IC Socket Kit, MITS 680b mod documentation, Universal Adaptor Kit (converts CIS-30+ for use with any computer).

MIKBUG® Motorola, Inc.



For your S-100 computer — the CI-812

- Both cassette and data terminal interfacing on one S-100 bus PC board.
- Interfaces *two* recorders. Record and playback circuits are independent.
- Select 30, 60, 120, or 240 bytes per second cassette interfacing, 110 to 9600 baud data terminal interfacing.
- KC-Standard/Bi-Phase-M (double frequency) encoded cassette data. Dependable self-clocking operation.
- Optional firmware (2708 EPROM) Operating System available.
- Prices: kit, $99.95; assembled, $129.95.

Prices include a comprehensive instruction manual. In addition to the EPROM Operating System, a Test Cassette, Remote Control Kit (for program control of recorders), and an IC Socket Kit are also available.

For your data storage — Pilon-30™ data cassettes

- Orders-of-magnitude improvement in data integrity over ordinary audio cassettes.
- Pilon-coated pressure pad eliminates lint-producing felt pad of standard audio cassettes.
- Smooth pilon coating minimizes erratic tape motion.
- Foam pad spring is energy absorbing. Superior to leaf spring mounted pad which tends to oscillate and cause flutter.
- Five-screw case design virtually precludes deformation during assembly.
- Price: $2.49.

PERCOM™ products may be purchased from home computer dealers nationwide, or may be ordered direct from the factory.*

*Texas residents must include an additional 5% for factory orders. MC & Visa cards honored.

PERCOM™ *'peripherals for personal computing'*

PERCOM

PERCOM DATA COMPANY, INC.
DEPT. I
318 BARNES • GARLAND, TEXAS 75042
Phone: (214) 272-3421

Shipping time is another objection. We ship all products within a week unless our suppliers have shorted us on parts. On the one product where we did experience a significant delay, we gave all customers a chance for a full refund after 30 days. For those who retained their orders, we offered a $5.00 rebate on a $50 item and shipped the product within the time period specified in our rebate offer. And on that product our initial supplier accounted for 100% of the delay. For items not in stock, we offer immediate refunds, usually returning the customer's check on the same day.

Manufacturers are not all rip off artists. And we're not all huge insensitive conglomerates. In fact, when we had a demonstration Star Wars game program more advanced that any other video game on the market at the West Coast Faire we responded to customer requests by supplying the 14K program for a $5.00 copy charge. I'd like to see the computer stores match that offer. And we didn't even intend to ever sell it.

Call 'em as you see them, but let's call 'em fair.

Barry Mittan, Vice President
Objective Design, Inc.

Mr. Mittan, your points are well taken, as were Mr. Osborne's. Although we do not like to use the pages of the magazine to carry on a meaningless diatribe, we feel it is important to provide a forum to parties presenting valid information. Therefore we have printed your letter in full.

Dear Editor:

This is a modification of "Look" program by Kenyon Swartout from the May 1978 INTERFACE AGE Magazine, pages 166-168. This change will allow output of unknown bytes in the vicinity of specified bytes by use of optional "XX" entry. Output will be arranged in columns of addresses and data. All line numbers from the original text not listed here remain unchanged. Entering this listing into the original text will change or delete instructions as required.

Richard H. Smith
Duluth, MN

```
0381         LDA   BEEN        ;HAVE WE BEEN TO LK2 YET
0382         ORA   A           ;WE DONT WANT THE HEX DATA ROUTINE
0383         MOV   A,B         ;IF WEVE BEEN TO LK2
0384         JZ    HEX         ;NOT YET? THEN JUMP TO HEX
0385         CPI   'X'         ;IF WE HAVE BEEN, THEN WE
0386         JZ    LK1         ;DONT ALLOW 'X' INPUTS GO BACK IF 'X'
0390 CONT    CALL  CHOUT       ;SAME AS BEFORE EXCEPT LABEL ADDED

0420 NOND    CALL  CHIN        ;SAME AS ORGINAL EXCEPT LABEL ADDED
0422         CPI   'X'         ;WE DONT ALLOW 'X' INPUTS NOW
0425         JZ    NOND        ;GO BACK IF 'X'

0540 CONT1   INX   D           ;AS BEFORE WITH A LABEL NOW

0593 HEX     CPI   'X'         ;IS INPUT AN 'X'
0594         JNZ   HDATA       ;NO, MUST BE HEX DATA SO JUMP
0595         STAX  D           ;YES, THEN SET FLAG BYTE HIGH WITH CHAR IN AC
0596         INX   D
0597         CALL  CHOUT       ;ECHO CHAR. TO TERMINAL
0598         CALL  CHIN        ;LETS REQUIRE TWO ENTRIES SO SAME AS HEX DAT/
0599         MVI   B,'X'       ;WE'LL MAKE IT AN 'X' AS THAT IS ALL THATS OK
0600         CALL  CHOUT       ;ECHO AN 'X' REGARDLESS OF INPUT
0601         MVI   B,' '       ;OUTPUT A SPACE SEPARATOR HERE
0602         CALL  CHOUT
0603         JMP   CONT1       ;CONT LK1 BEYOND CALL TO AHEX DONT NEED IT
0604 HDATA   XRA   A           ;ZERO ACC. AND USE IT TO
```

```
0605          STAX  D              ;ZERO FLAG BYTE TO INDICATE HEX DATA FOLLOWS
0606          INX   D
0607          JMP   CONT           ;CONT LK1 NEAR WHERE WE LEFT
0608 LK2      POP   H              ;SAME AS BEFORE. HAD TO CHANGE LINE NUMBER

0625          STA   BEEN           ;THIS BYTE IS ZEROED AT START SET HIGH BY LK2

0795          CALL  CRLF           ;SOME CRLF'S BEFORE LISTING DATA
0797          CALL  CRLF

0872          STA   FLAG           ;SAVE FLAG BYTE FOR LATER
0874          LDAX  D              ;HEX DATA. IGNORE IF PRECEEDING FLAG IS HIGH
0876          PUSH  PSW            ;SAVE IN STACK IN CASE WE WANT IT

0902          LDA   FLAG           ;GET FLAG BYTE BACK SO WE
0904          ORA   A              ;CAN SEE IF IT IS HIGH
0906          JNZ   SKIP           ;HIGH? THEN IGNORE HEX DATA IN STACK JUMP SK

0962 SKIP     POP   PSW            ;GET RID OF HEX DATA IN STACK DONT NEED IT

1022          LDAX  D              ;FETCH FLAG BYTE
1024          INX   D
1026          ORA   A              ;SEE IF ITS HIGH
1028          JNZ   LOOK3          ;HIGH? HEX DATA FOLLOWING IS A "FORCED" MATCH

1100 ;DELETED INSTRUCTION. STACK FIX BY POP PSW IN LINE 1810
1110 ;DELETED INSTRUCTION. CRLF DETERMINED BY PPOS BYTE
1120 PRT      LHLD  SA             ;SAME AS ORIGINAL. JUST MOVED LABEL DOWN

1250          LHLD  SA             ;START ADDRESS OF FOUND MATCHED DATA
1255 ;NOW WE CAN OUTPUT HEX DATA WE FOUND THERE

1455          CALL  POS            ;COLUMNIZE DATA DISPLAYED

1490          ASC   'ENTER "XX" FOR UNKOWN BYTES: HEX BYTES?

1800          RNZ                  ;RETURN IF NOT DONE
1810          POP   PSW            ;OTHERWISE GET RID OF RETURN ADR IN STACK
1815          JMP   PRT            ;JUMP TO PRINT ROUTINE

2132          CPI   'X'            ;IS IT AN "X"
2134          RZ                   ;RETURN IF "X"

2210 CHOUT    LDA   PPOS           ;WHAT IS OUR PRINT POSITION
2212          INR   A
2214          STA   PPOS           ;INCREASE IT BY ONE AND PUT BACK
2215          IN    0              ;SAME AS BEFORE. HAD TO CHANGE LINE NUMBER

2341 POS      LDA   PPOS           ;WHAT IS PRINT POSITION
2343          CPI   40             ;IF OVER 40 PLACES (FOR 64 CHAR. DISPLAY)
2344          JP    CRLF           ;THEN WE NEED A CRLF NOW
2345          MVI   B,' '          ;OTHERWISE WE WILL
2346          CALL  CHOUT          ;JUST OUTPUT 3 SPACES
2347          CALL  CHOUT
2348          JMP   CHOUT

3190          MVI   B,34           ;WE NEED TO ZERO 12 MORE BYTES

3290 BYTE     DS    19             ;9 FLAG BYTES, 9 HEX BYTES AND ONE SLOP

3332 FLAG     DS    1              ;TEMPORARY FLAG BYTE STORAGE
3334 BEEN     DS    1              ;ZERO UNTIL WEVE BEEN TO LK2
3335 PPOS     DS    1              ;PRINT POSITION. ZEROED BY CRLF

                                                                        001
```

Dear Editor:

I must point out an error in my article, "A 24 Line Display for the Heath H9" appearing in the June 1978 issue of INTERFACE AGE.

Under item 3 of the character generator board modification list, "Jump U221-2 to U219-11" should instead read "Jump 218-2 to U219-11". The article has been otherwise accurately reproduced.

Stephen Sama
4006 Berrywood Drive
Seaford, NY 11783

Steve, we appreciate the update. We have had several phone calls on this.

Dear Editor:

Output from the FMIAPS (6800) program in the May, 1978 issue was erratic when I tried it. Investigation of the problem indicated control characters in the output and a format different than the MSI (Uiterwyk) BASIC TSAVE/TLOAD tape format.

The enclosed program listing creates a tape in the correct format and only print characters or intended double sequence control characters are in the output tape. A sample listing of the first program on the record is enclosed. This format is reliable and the correct format produces a readable input with the CR/LF.

Incidentally, the backside of my record — the datafile, was not usable because of a deep, wide scratch. This appeared in two other copies I examined, and it appears to me to possibly be caused in binding of the magazine since they all appeared identical.

B.F. Sedlack
Albuquerque, NM

```
S11301007E014301020304050D0AA042FFFF000122
S11301100237023700000D0A0000524541445920BD
S1130120464F5220494E505554041300D0A0000052B4
S11301304541445920464F52204F555450555404 7C
S113014007EE07EBE010AFE0110FF0112CE01168D73
S1130150EF8D39B1010826F9B601098D2CFE011283
S113016086118D258D26B101062724B1010326F4BD
S11301708D1AB1010426F9FE01128D10A70008B1F1
S113018001 0526F6FF011220DB7EE1D17EE1ACA75A
S113019000086148DF4CE012A8DA68DF0B1010826B7
S11301A0F9B601098DE386128DDFFE01108D5D7FA6
S11301B0010EA600B101062741B1010527297D01E1
S11301C00E2608B601048DC173010EA60008811025
S11301D02609A60008843F8A80200881202DD38127
S11301E05F2ECF8DA420CB08B601088D9CB60109E3
S11301F08D9786158D938D3420B5B601058D8A862D
S1130200138D868D0786148D807EE0D08D098D0731
S113021008D058D038D013937FF0114F6010FFE01A1
S11302200C0926FD5A26FAFE01143339C60486FF4A
S10A0230BD01895A26F839CB59
```

We had a binding problem and consequently replaced a lot of records.

Dear Editor:

I very much enjoyed the article on the H8 computer system, as presented in the June 1978 issue of INTERFACE AGE. However, I did catch one small error on page 136 concerning the software included with the system.

Mr. Arnold mentions that the text editor (TED-8) "...does not appear to have the capability of processing text in upper and lower case simultaneously." Now, I was one of the first to pick up my H8 when it was available in my area, and, unless Mr. Arnold managed to get a hold of an extremely old copy (if such existed) of TED-8, the statement he made is not correct.

Any time during the running of the TED-8, the user can press the 'L' key, and the editor will print out the "LOWER CASE (Y OR N)?" prompt for the user.

However, this slip was minor, and I did enjoy the article very much. How about seeing some BH BASIC programs, or assembly listing for the H8 in your software section?

Craig A. Pearce
Berwyn, IL

Thanks, Craig.

Dear Editor:

I look forward to receiving your magazine every month and thoroughly enjoy the software you make available. Okay, that's out of the way.

I am a hardware-type, having worked with digital for several years prior to buying my Imsai. My learning process includes finding software, discovering how it works, and modifying it, and eventually being able to write my own. I entered your "Piranha" game of a few months back, and enjoy it. What I did not enjoy was the time wasted trying to figure out what in the listing were "F" and not "E". (Strikeovers due to my typing, of course.) Now I wanted to try to load your Floppy ROM™ Mailing List, began entering the 8080 version of the software to do this, and found that whoever printed the listing had some terrible problem, either hardware or software, but which made several portions of it unusable. If the reader must rewrite the program, why bother with it? Then I went to the "LOOK" program. Very usable software, but just try to tell the "8" from the "B". These minimum matrix dot printers are a curse to the reader at best, but when reduced in size, or in the earlier case, in no way proofread, they become almost useless, like the first example, a TTY out of alignment. Orchids for the software, a rafflesia for your way of printing it.

Dave Powell
Lexington, KY

What can we say. We can only try.

Dear Editor:

I have converted the Memory Catalog Program by Jim Baumgardt in the May 1978 issue of INTERFACE AGE to run on a Poly-88. My computer uses memory mapped video out-

put instead of a conventional terminal. I thought other readers might be interested in this conversion. The conversion uses some of the routines in the POLY ROM monitor and is therefore somewhat shorter than the original program. Make these changes:

These lines are not needed — 10 through 220, 240 through 280, 650 through 660, 720 through 970, and 1190 through 1270.

Add these lines (you don't need the line numbers with the new POLY assembler, but they'll help you figure out where to put the lines).

```
140 CLEAR   EQU   0392H
150 CROUT   EQU   038DH
160 DEOUT   EQU   03D1H
170 HEXOT   EQU   03D6H
180 OUT     EQU   0C24H
190 WHO     EQU   0C20H
```

Replace line 0230 with ORG ALL CLEAR.

Change the following lines as indicated:

```
370             MVI    A,02DH
530             MVI    A,020H
620 RET         CALL   CROUT
630             CALL   WHO
640             JMP    OH
```

After the indicated changes are made and the program is run, the screen will clear, the addresses of good RAM memory will appear, and a carriage return and line feed will be output to the screen. Touching any key at that point will return you to the POLY monitor as if the POLY had just been turned on.

I assembled this program using the excellent new POLY version GO2 Assembler/Editor. The editor is a true character oriented editor which is great for writing assembly programs, but it's also nice for use as a primitive word processor. If you have not seen GO2, then try to get someone to demonstrate it to you. It will be worth your while.

David L. Johnson
Prince George, VA

Thanks, David, we are sure many readers will enjoy your conversion.

Dear Editor:

Thanks for sending me the missing issue; can't beat that for quick service. Some magazines ignore reader subscription complaints, or at least don't rectify the problem fully.

Regarding my program published in the March issue (A 6800 Relocator); that *is* the complete program, although due to my delay in submitting some sort of a flow chart, a mix-up occurred. I decided to shorten the addition and subtraction subroutines, which put the end point at $01CA instead of $0208. This change was included in the revised textual part of the article, however, the previous text was used by INTERFACE AGE and that, understandable as it was from an editorial standpoint, evidently led to some confused readers. By comparing the listing with the modified Warnier/Orr diagram it can be seen that *nothing*

is really missing.

One feature I will emphasize again: since no relocation process can 'know" the purpose of X-Register Immediate operands, they will be changed along with other extended addresses, wanted or not, so it might pay to check programs after relocation. The following will relocate my program to start at $3110: after the X prompt, type 011001CA3110N, then a carriage return. The 'N' can of course be any other character except a 'T', which is reserved for moving textual portions of a program.

A mild gripe is now offered: it seems as if the hobbyist is about to drop by the wayside in favor of the small businessman. Perhaps one of the magazines (maybe INTERFACE AGE!) should become, gradually, a trade publication for the micros and the others could then continue business as usual for the Computerniks.

I really like Fountainhead; read it first each month.

Neal Champion
Prescott, AZ

Neal, we appreciate the kind words. Thanks for clearing up the problem.

Dear Editor:

I think your Floppy ROM is a highly commendable idea; although I found it personally disappointing. When the first one with BASIC for the 6800 came out, I eagerly anticipated a similar one for the 8080. However, you seemed to have assumed that most users already have BASIC. I may easily be in a minority without it, but certainly most people do not (yet) have the extensive memory and peripherals required by the subsequent releases. Could we have perhaps an issue with less stringent requirements, and hence a wider appeal?

Some random (very biased) thoughts on your magazine:

Sorely missed: Those excellent hardware articles, e.g. the various articles on specific chips, and the series on interfacing to the S100 bus.

Most interesting article: The description of the "Artificial Intelligence" program, whereby "voters" acting individually on seemingly meaningless and random criteria could collectively produce meaningful results.

Least interesting articles: All those 'business' articles. The macro computer field soon reduced 99% of computer-related activities to a tremendously boring level. Is the personal computer domain to suffer a similar fate?

Overall: very good. I brought some issues into work to show people. Needless to say they disappeared when I wasn't looking.

R. G. Sharman
Ottawa, Ontario, Canada

Mr. Sharman, we try to get a good broad spectrum of articles in each month, but we can't satisfy everyone. The new products directory is our most read item so we keep it.

DATA GENERAL ENTERS 16-BIT SINGLE-BOARD COMPUTER MARKET

A single-board computer offering minicomputer performance and systems capability combined with microcomputer board technology and economy was introduced by Data General Corporation. The introduction represents the first fully-featured 16-bit single-board computer to be offered by a major minicomputer manufacturer.

The MicroNOVA Board Computer (MBC/1) includes a 16-bit microNOVA CPU, 2 K-bytes of static RAM, sockets for up to 4 K-bytes of PROM memory, an asynchronous communications interface, and a 32-line digital input/output port — all on a single 7.5" x 9.5" board. The CPU provides full NOVA® architecture, hardware stack and frame pointer, 16-bit hardware multiply and divide, real-time clock, hidden memory refresh, data channel (DMA), and 16-level priority interrupt.

Software for the MBC/1 includes a multitasking support package (MBC/M) that provides an emulator for program development under all Data General operating systems and a monitor for program execution on MBC/1. Optional on-board ROM console debug and self-test diagnostics are also available. These features permit software development on larger minicomputer systems.

The CPU of the MBC/1 is a 16-bit microNOVA microprocessor. Two types of memory are offered on the board: 2 K-bytes of static RAM, and sockets for up to 4 K-bytes of PROM for implementing programs in 512-word (1 K-byte) increments. The memory bus extends off the board to allow the user to add additional memory. Both the RAM and PROM memories are compatible with the microNOVA product line.

The MBC/1's digital I/O interface provides 16 input lines, an external interrupt line, 16 output lines, a data strobe, and a system reset line. The asynchronous interface provides full duplex communication with an asynchronous terminal or modem via either a 20 mA current loop or EIA RS 232/C lines with speeds from 110 to 9600 baud. Line speed, data bits, parity, and stop bits are jumper selectable. The I/O bus extends off the board for enhanced expandibility.

The MBC/1 can also be used in data acquisition systems, such as those for meteorological and air pollution monitoring and demand data collection at unattended locations by equipment that is both small in size and reliable. The MBC/1 can be added to other control equipment without major repackaging or system redesign in a variety of industrial processes that need computing capability at machine level locations. Its compact 16-line flat ribbon I/O bus simplifies cable routing and cuts cable costs. The differential drive of the I/O bus signals allows I/O cables to run alongside electrical noise sources without detrimental effects to the transmitted data.

NOVA is a registered trademark and microNOVA is a trademark of Data General Corporation, Westboro, Massachusetts.

PET USERS GROUP

A users group and newsletter is being started for PET owners. To encourage an exchange of information and software, we are publishing "TRANSACTION," a bi-monthly newsletter.

Anyone interested in contributing ideas or articles, or who would like to receive the newsletter can write to: TRANSACTION, P.O. Box 461, Philipsburg, PA 16866. A one year subscription is $3.

APPLE USER'S GROUP

The New Jersey Apple Users group has been recently formed. It meets at the Computer Lab of New Jersey the first Friday of each month.

For further information contact Dan Fischler at the Computer Lab, 141 Route 46, Budd Lake, NJ 07828.

GRT TO PRODUCE PROGRAMMED CASSETTES FOR HOME COMPUTERS

A library of programmed cassette tapes for personal home computers will be introduced by GRT Corporation. The first tapes will be available in 60 days and will be marketed through personal computer retailers and department stores nationally.

Initial offerings will include the following programs: home finance, including checkbook balancing and loan amortization; stock option tracking; cash flow analysis; diet assistance and medical biorhythms; and several computer games including blackjack and bridge instruction.

For more information contact Carter Elliott, GRT Corporation, Sunnyvale, California, (800) 662-9810. From out of state call (800) 538-1770.

EXCHANGING SOFTWARE

The Software Exchange is a new publication devoted to promoting the exchange of software in the small computer marketplace. Reviews, articles and advertisements will focus on information useful to people who are interested in putting their computers to work.

The Software Exchange is a bimonthly magazine available soon at computer stores for $1.50 per issue and by subscription for $8 per year (six issues).

People with software to sell or trade, and those looking for software to buy, can place classified advertisements in The Software Exchange for a nominal $2 fee. If our Classified Form is not available from the local computer store or club, the following information should be included with an indication if it is for sale or wanted:

1. Application (Business/Finance, Word Processing, Sciences/Engineering, Statistics/Mathematics, Home/Personal, Games/Entertainment, Systems, or Miscellaneous). The ad will be placed in one of these sections.
2. Description of Program
3. Hardware/Software Requirements
4. Materials/Media/Pricing
5. Source or Destination (name and address)

SMALL COMPUTER USERS' GROUP FORMED

Reacting to "a bewildering array of new computing alternatives," a users' group has been formed to provide a source of "unbiased, user-oriented information" on mini and micro computers for business applications.

The new Association of Small Computer Users (ASCU) plans to provide members with selected publications at reduced cost, a bimonthly newsletter and information exchange, and benchmark comparisons of competing small computer systems, according to newly-elected ASCU President Hillel Segal.

Membership fees will be $25 per year for individual current or prospective users of small computers, and will include a number of periodicals and reports. Membership information may be obtained from The Association of Small Computer Users, 75 Manhattan Drive, Boulder, CO 80303.

Xitan inc.

SOFTWARE

XITAN SOFTWARE IS WIDELY RECOGNIZED AS THE **BEST** IN THE **MICRO-COMPUTER** INDUSTRY. IT'S AVAILABLE IN **PAPER TAPE, CASSETTE** AND **FLOPPY DISK** VERSIONS FOR A VARIETY OF HARDWARE CONFIGURATIONS!

DISK BASIC, the best most powerful interpreter available today. Loaded with functions and features to provide a most versatile system for applications in business, science, engineering or your personal computing pleasure. Look! All devices may be handled in single byte modes; sequentially, randomly or in update modes. Allows use of a PRIVACY statement; global editing and much, much more. Available in CP/M™ version.

Z-BUG, a highly flexible, dynamic, reliable, disk based programmed de-bugging utility. With user controlled data formatting, six commands to manipulate memory, provision to display the next instruction before executing and this is only the start! This piece of software could well be one of the most important "tools" you'll ever use. Available in CP/M™ version.

LINKER, combines routines written in different languages, no need to recompile an entire program after debugging one or several modules, use routines from other disk files or libraries. Each separately compiled module, after linking, can access code or data defined in other modules. Available in CP/M™ version.

ZAPPLE MONITOR, occupying 2K of memory, provides unparalleled power in a serial I/O environment, supporting 4 logical I/O devices. It also supports relocatable, absolute and binary files, 26 commands and an extensive debugging capability. Available on Paper Tape.

Z-TEL For the programmer manipulating large source files or an author preparing a text for a book, both find big machine text editing facilities available. Capable of moving large blocks of text, programable to allow expression evaluation, iteration, conditional branching and subroutine calls. Available on Paper Tape, XITAN Cassette and in CP/M™ version.

TEXT OUTPUT PROCESSOR. Using Z-TEL, special text format control words are entered into your file, processed by TOP and output to the printer is formatted as you ordered with automatic headings, page numbering, left and right justified margins, indentation, centering and more. Available on Paper Tape, XITAN Cassette and CP/M™ versions.

MACRO ASSEMBLER The best resident assembler available. Generates either relocatable or absolute object codes in two formats. Extremely powerful macro definition facility and allows programmer control over assembly parameters and listing format. Uses a unique opcode set compatible with INTEL 8080 mnemonics, inter-module linkage capability, library search and is available on Paper Tape, XITAN Cassette and CP/M™ version.

All XITAN software includes full documentation and user manuals.

ZAPPLE BASIC, with 12 operators, 24 functions and 44 commands, is a powerful interpreter exceeding the capabilities of other 8K BASIC's. Add four more functions: EDIT, LVAR, RENUMBER, TRACE for a total of 28 functions and you have the makings for a system to provide for an ever growing number of capabilities. Available on Paper Tape, and XITAN Cassette only.

SUPER BASIC adds more functions and commands to the features of ZAPPLE BASIC, for easy program entry, program chaining, formatting output and improved string manipulation. Eleven digit accuracy for business applications. Available on Paper Tape, XITAN Cassette and CP/M™ versions available with direct interface to disk files for serial I/O.

VDB BASIC adds the special commands to SUPER BASIC to control the VIDEO DISPLAY BOARD. Write directly to the VDB; plot a point or line; clear or read; position a cursor or allow multiple VDB's.

MICRO-SEED, the first and only Data Base Management System for a microcomputer, is an implementation of CODASYL standard for data base systems. Makes programs and data independent of each other. Simplifies the integration of many applications into the same database.

MICRO-SEED is only available in a CP/M™ version and requires XITAN FORTRAN IV for its use. Developed for XITAN by International Data Base Management Systems, Inc.

FORTRAN IV, developed for Xitan, by Small System Services, Inc., is an ANSI standard FORTRAN IV with a broad variety of extensions, far above any other micro based Fortran. It is supplied with an extensive library of support routines. Available only in a CP/M™ version.

™CP/M is a registered trademark of Digital Research.

XITAN, INC.
Corporate Headquarters:
1101-H STATE ROAD,
PRINCETON, NEW JERSEY 08540

Software Division:
639 MASSACHUSETTS AVE., SUITE 214,
CAMBRIDGE, MASS. 02139

XITAN, INC. 1101-H State Road, Princeton, N.J. 08540

Please send me information:

- ☐ Disk Basic
- ☐ Linker
- ☐ Z-Bug
- ☐ Fortran IV
- ☐ Micro-Seed
- ☐ The complete line of Hardware & Software plus Price List.

I AM
- ☐ Hobbyist
- ☐ Small Business User
- ☐ Educator User
- ☐ Engineering User
- ☐ Other ____________

NAME ____________

ADDRESS ____________

CITY ____________

STATE ____________ ZIP ____________

My Dealer's Name ____________

CIRCLE INQUIRY NO. 56

All software now available on North Star CP/M™ and Micropolis CP/M™.

WESCON THEME CENTERS ON MICROPROCESSOR APPLICATIONS AND AWARDS PROGRAM

The revolutionary impact on electronics of the microprocessor has made the tiny chip the central theme of the giant Wescon/78 high-technology convention and exhibition and spawned an awards program saluting applications of the device.

Home, industry, automotive, energy and games and toys are among the eight categories selected for special microprocessing awards as part of the theme program of Wescon, September 12-14, at the Los Angeles Convention Center.

LOGIC ANALYZER SEMINAR

To acquaint engineers with opportunities for logic analysis in digital systems, a one-day seminar on the use of logic analyzers is now available free of charge through Tektronix Field Engineering Offices.

The seminar examines the subject of digital measurement during the audio-visual portion, describing the problems of digital analysis and the capabilities of analyzer instruments in the areas of multiple parallel input, negative time triggering, memory, word recognition and display formatting.

For further information on how groups or individuals can arrange for the seminar, contact the area Tektronix Field Engineering Office or Tektronix, P.O. Box 500, Beaverton, OR 97077, (503) 644-0161.

SCORE

A conference series designed for both prospective and established small business owners will highlight the 4th Annual California Business and Industry show to be held at the Anaheim Convention Center, September 19-21, 1978.

Co-sponsored by the U.S. Small Business Administration and SCORE of Orange County, the series will focus on successful marketing techniques and sound financial planning.

Aimed toward the end user market, the diversified California Business and Industry show will feature a variety of products and equipment including small business computers and other computer related products and services oriented toward medium to small businesses. Further information regarding the show and conferences may be obtained by telephoning (714) 558-0846.

2ND ANNUAL PERSONAL AND BUSINESS SMALL COMPUTER SHOW

The 2nd Annual Personal & Business Small Computer Show is scheduled for September 15-17, 1978 in the New York Coliseum.

The upcoming event promises to have an attendance goal of 20,000 announced by show management.

Under development is a 7-phase lecture series covering introductory material, software, small business and professional applications, recreational and household applications, and educational applications, in addition to a computer retailing workshop.

Admission to the show is $5.00 per day at the door. Non-refundable two-day admissions at $9.00 and three-day admissions at $13.00 may be purchased in advance only. Mail orders will be taken up to September 4 by Personal & Business Small Computer Show, 78 East 56 Street, New York, New York 10022.

PROJECT MANAGEMENT FOR COMPUTER SYSTEMS

The increasing importance of specialized skills for the administration of complex computer projects is explored in "Project Management for Computer Systems," a 3-day seminar presented by The University of Chicago Center for Continuing Education in four cities around the country:

October 24-26, 1978 New York
December 11-13, 1978 Chicago
January 22-24, 1979 San Francisco
March 19-21, 1979 Atlanta

This practical, 3-day seminar examines and illustrates techniques for planning, implementing, installing and controlling computer projects. The program emphasizes the management of costs, schedules and quality.

For a detailed brochure and registration information, contact Heidi E. Kaplan, Dept. 20NR, New York Management Center, 360 Lexington Ave., New York, NY 10017, (212) 953-7262.

VANTAGE ANNOUNCES NEW MONTHLY NEWSLETTER COVERING THE PERSONAL COMPUTING INDUSTRY

The Personal Computing Industry Report provides a single source for news, market analysis, produce reviews, company profiles, technology forecasts and applications case studies. PCIR provides full coverage of the personal computing industry in all five major applications areas: Business, Professional, Education, Hobby and Home.

The *Personal Computing Industry Report* is available from Vantage Research at an annual subscription price of $195. The firm is located at 770 Welch Rd., Palo Alto, CA 94304.

CRAMER ELECTRONICS ANNOUNCES COURSE SCHEDULE

Cramer Electronics, Inc. has announced the schedule for Cramer

University, a five-month schedule of microcomputer courses to be held at their Technical Training Center in Newton, Massachusetts.

This series of seminars is designed to assist technical people, engineers and designers who need to be kept updated on the latest microcomputer developments and how these new systems can benefit their applications. The seminars are offered free on a limited attendance basis. Those wishing course outlines and schedules or wanting to make a reservation may call their local Cramer division or (617) 969-7700, Ext. 303.

EIA/DPD CENTRAL ANNOUNCES 78-79 MEETING PLANS

The Central Region Planning Committee of the Electronic Industries Association/Distributor Products Division has proposed a series of meetings for the 1978-79 program year.

Among the topics for the general sessions of the DPD meetings will be seminars on "Managing Your Sales," "Managing Your Business and Family Life," "Managing Distributor Relations," "Managing Your Manufacturer-Rep. Relations," "Managing the Sale of Electronic Products to Non-Traditional Markets."

Meetings will be held on September 19th, November 14th, December 5th, January 6th, February 22nd, March 20th, and June 12th. There will be no Central Region Meeting in October or April because of the national EIA meetings, and none in May because of the NEWCOM Show.

Information about DPD programs is available from the Central Region Office at 222 So. Riverside Plaza, Chicago, IL 60606, (312) 648-1600 or from the National Office, 2001 Eye St., NW, Washington, D.C. 20006, (202) 457-4930.

THREE-DAY SEMINAR ON "AUTOMATION OF MANUFACTURING OPERATIONS"

New York University's School of Continuing Education has announced additional dates for its 3-day seminar "Automation of Manufacturing Operations" to be held October 18-20, 1978 in Toronto, December 11-13, 1978 in New York City, April 23-25, 1979 in San Francisco, and June 11-13, 1979 in Chicago. This three-day seminar covers the theory and practice of automation and its interrelationships with manufacturing processes, materials, labor management and economics.

For a detailed brochure and registration information, contact: Heidi E. Kaplan, Dept. 20 NR, New York Management Center, 360 Lexington Ave., New York, NY 10017, (212) 953-7262.

NEW FIRM PROMOTES MICROS

Man Computer Systems, Inc. has recently been formed to help investors and consumers profit from new advances in computers.

Man-Computer Systems, Inc. is a New York based consulting and publishing firm dedicated to promoting the use of microcomputers for personal computing and small business applications through publications, software development, and microcomputer systems design.

For further information contact Dr. Jerry Felsen, President, Man-Computer Systems, Inc., 84-13 168th St., Jamaica, NY 11432, (212) 739-4242.

WESTERN DESIGN ENGINEERING SHOW

The first major regional event for the design engineering field — the Western Design Engineering Show — has passed the 100-mark in the number of companies which will participate as exhibitors.

The show, which will take place at the Anaheim Convention Center, Anaheim, California, December 5-7, 1978, will be accompanied by a 3-day Design Engineering Conference, sponsored by the American Society of Mechanical Engineers.

In addition to the electronic aspects of product design, the Western show and conference will provide new approaches to mechanical components, power transmission equipment, electrical components, materials, hydraulic and pneumatic components, fasteners and joining materials, shapes and forms, finishes and coatings, and engineering equipment and services.

A guide to the show and a complete conference program are in preparation and will be available by writing to Clapp & Poliak, Inc., 245 Park Ave., New York, NY 10017.

CALL FOR PARTICIPATION ISSUED FOR 1979 NCC

A Call for Participation has been issued for the 1979 National Computer Conference to be held next June 4-7 in New York City. All individuals in the information processing field, including both computer specialists and data processing users, are invited to write a paper, propose a technical or panel session, volunteer to be a panelist, send ideas for topics, or suggest special activities, demonstrations, or conference features which they would like to see included in NCC '79. Deadline for all submissions is November 1, 1978. To obtain a copy of the official Call for Participation contact AFIPS, 210 Summit Avenue, Montvale, NJ 07645.

All suggestions, proposals, and papers should be sent to the pro-

gram chairman: Dr. Richard E. Merwin, Box 32222, Washington, D.C. 20007. Guidelines for participation in the conference program are contained in the official Call for Participation available from AFIPS headquarters.

OEM REP/DISTRIBUTOR LIST LAUNCHED BY MICROPOLIS

Micropolis Corporation has signed manufacturer's representatives/distributors in 17 states.

New Micropolis representatives with multiple sales offices include J&J Associates in New England, Group III Electronics in California, K/A Electronics in Texas, Data Electronics in the Midwest, and Par Associates in Colorado, Utah, Arizona and New Mexico.

Services offered by these firms are tailored to the two-tiered nature of our OEM marketplace. To serve the large volume manufacturer who integrates floppy disk modules into end-systems, the representatives will provide technical support with factory backup and direct shipment. For the system integrator — who buys all hardware components and adds value through configuration and specialized software — the reps will serve as a stocking distributor.

SPECIAL MANAGEMENT SESSION SET FOR DPMA'S NEW ORLEANS '78

A Major highlight of the International Conference and Business Exposition to be held by Data Processing Management Association (DPMA) in New Orleans October 29-November 1 will be a special management seminar featuring corporate executives outlining opportunities for data processing managers in reaching corporate-level positions.

The conference will be held in the New Orleans Hilton Hotel and the special session will be presented the morning of October 31.

It will offer more than 40 topical seminars on EDP technology and management techniques, presented by experts in their field of specialization.

For conference information, contact Carol Harte, conference coordinator, DPMA International, 505 Busse Highway, Park Ridge, IL 60068, (312) 825-8124.

QUME LINES UP THIRD PARTY MAINTENANCE FOR SPRING 5™ DAISYWHEEL PRINTER TERMINAL

Qume Corporation has contracted with Sorbus Inc. to provide maintenance on its recently announced Sprint 5 daisywheel printing terminals.

The agreement provides for initial service in major metropolitan cities and will expand to nationwide coverage. Qume will supply parts and training materials enabling Sprint 5 customers to contract directly with Sorbus for maintenance.

The Sprint 5 terminal is primarily aimed at computer industry printer applications, which in the past have lacked the high print quality Qume Daisywheel Printers provide. The Sprint 5 comes in Keyboard Send Receive (KSR) and Receive Only (RO) versions.

"CONTROL OF MATERIAL FLOW" SEMINAR SPONSORED BY THE UNIVERSITY OF CHICAGO

Effective management of a company's substantial investment in materials is the focus of "Control of Material Flow," a 3-day seminar sponsored by The University of Chicago Center for Continuing Education in three cities around the United States: October 16-18, 1978 Chicago; December 12-14, 1978 New York; April 4-6, 1979 San Francisco.

This practical seminar examines the role of efficient management systems in the flow of raw, in-process and finished materials and inventories. An integrated approach for raw materials purchasing through finished product delivery is emphasized.

For a detailed brochure and registration information contact Heidi E. Kaplan, Dept. 20 NR, New York Management Center, 360 Lexington Ave., New York, NY 10017, (212) 953-7262.

SEMINARS SET FOR IMPROVING PRODUCTIVITY AT CHECKOUTS

Seminars to help supermarkets and other mass-merchandising organizations improve checkout productivity will be held at Western Michigan University and at Auburn University in June and July.

The 2½-day seminars are designed for instructors and training personnel from both retail organizations and educational institutions. They are conducted by specialists from the participating universities and NCR Corporation.

Attendees participate in training sessions on electronic point-of-sale equipment. Also included in the seminar are training manuals and material and training on measuring productivity. The course costs $285 for the first participant and $250 for each additional person from the same firm or school.

CIRCLE INQUIRY NO. 48

CALENDAR

Sept 1 Crescent City Computer Club will hold its meeting at the University of New Orleans, Lakefront Campus at 8 P.M. Call Bob Latham at (504) 722-6321 for more details.

Sept 1 Microcomputer Information Group will meet at 7 P.M. at the Microcomputer Resource Center, 5150 Anton Dr., Rm. 212, Madison, WI 53719, (608) 274-8925. Len Lindsay, president.

Sept 2 Louisville Area Computer Club (LACE) will meet at the University of Louisville, Speed School Auditorium at 1 P.M. For details, write the club at 115 Edgemont Dr., New Alban, IN 47150.

Sept 2 South Central Kansas Amateur Computer Association, 9:00 A.M., Wichita Public Library, Wichita, KS. For further information call Chris Borger at (316) 265-1120 or Dave Rawson, 1825 Gary, Wichita, KS 67219, (316) 744-1629 for further details.

Sept 2 Oklahoma Computer Club will be meeting at the Belle Aisle Library at 10 A.M. Call Al Campbell at (405) 842-4933 for details.

Sept 2 Milwaukee Area Computer Club will meet at 1 P.M. at the Waukesha County Technical Institute, New Berlin, WI. Call (414) 246-6634 for further details.

Sept 2 Southern Nevada Personal Computing Society will meet at Clark County Community College, Las Vegas, NV at 12:00. The club also meets on the 3rd Saturday of the month. For further information write SNPCS, 1405 Lucille St., Las Vegas, NV 89101 or call (702) 642-0212.

Sept 3 The Computer Hobbyist Group will meet at 1 P.M. in the Green Center, Rm 2.530, of Univ. of Texas, Dallas. For details write to P.O. Box 11344, Grand Prairie, TX 75051.

Sept 4 Amateur Radio Research and Development Corp. (AMRAD) meets the first Monday of each month at 8 P.M. at the Patrick Henry Branch Library, 101 Maple Ave. E, Vienna, VA. for details write the club at 1524 Springvale Ave., McLean, VA 22101.

Sept 4 Minnesota Computer Society will meet at the Brown Institute, Room 51, 3123 E. Lake Street, Minneapolis, MN. For further information contact the Society at Box 35317, Minneapolis, MN 55435, Attn: Jean Rice.

Sept 5 Tidewater Computer Club will meet at the Electronic Computer Programming Institute, Janaf Office Bldg., Janaf Shopping Center in Norfolk. The club also meets on the 3rd Tuesday of the month. For details contact: C. Dawson Yeomans, Interface Chairman, 677 Lord Dunmore Dr., Virginia Beach, VA 23462.

Sept 6 New England Computer Society will meet in the cafeteria of the MITRE Corp. at 7:00 P.M. Located on Route 62 in Bedford, MA. Contact Dave Day at P.O. Box 198, Bedford, MA 01730, (603) 434-4239 for details.

Sept 6 Kitchener Waterloo Microcomputer Club will meet at the University of Waterloo, Room 3388, Engineering Bldg. #4, University Ave., Waterloo, Ontario, Canada at 7:30 P.M.

Sept 6 The Valley Computer Club will meet at 7 P.M. at the Harvard School located at 3700 Coldwater Canyon, Studio City, CA.

Sept 6 Columbus Computer Club will meet at the Center of Science and Industry at 7:30 P.M. For further information write c/o Fred Hatfield K8VDU, Computer Data Systems, 1372 Grandview Ave., Columbus, OH 43212, or call (614) 488-3347.

Sept 6 Lincoln Computer Club will hold its meeting at the South Branch Library located on 27th and South Sts. at 7 P.M. For more details write Hubert Paulson, Jr., 422 Dale Dr., Lincoln, NE 68510.

Sept 6 Great Gulf Coast Computer Club G^2C^3 in Mobile, Alabama, meets the first Wednesday of every odd month. For time and location of the meeting call (205) 478-1777.

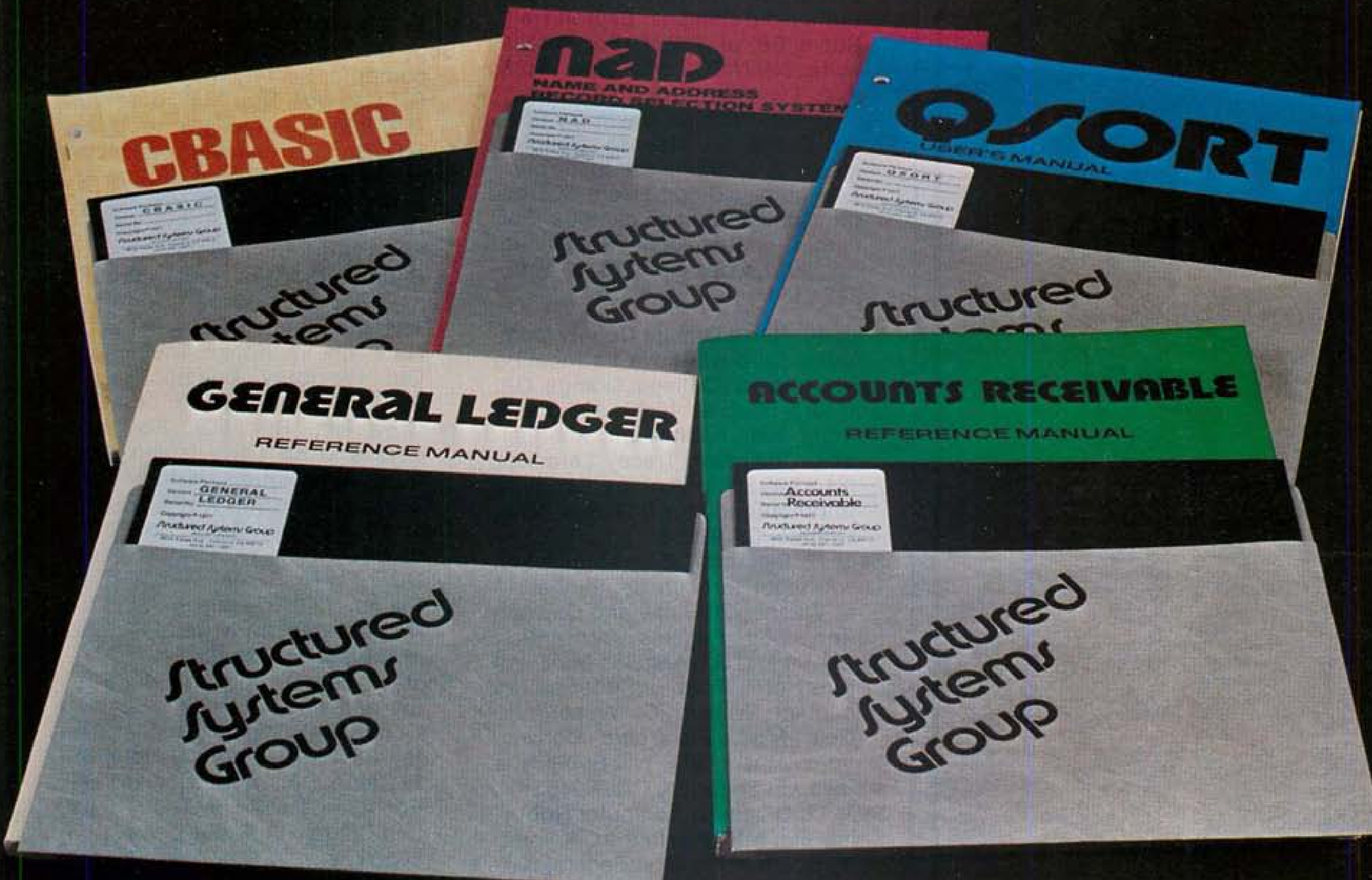
Sept 7 Bay Area Microprocessors Users Group (BAMUG) will meet in the Hayward ROC Center, 26316 Hesperian Blvd., Hayward, CA at 7:30 P.M. For further details write BAMUG, 1211 Santa Clara Avenue, Alameda, CA 94501.

Sept 7 Northwest Computer Society meets in the Pacific Science Center in Seattle, Room 200 at 7:30 P.M. The club also meets on the third Thursday of the month. For more details write NCCN, Box 4193, Seattle, WA 98055.

Sept 7 Microcomputer Users Group (MCG) will hold its meeting at the University of Minnesota, Electrical Eng. Rm. 115 at 7 P.M. The club meets every Thursday. For more information write MCG, Dept. of Elec. Eng., 123 Church St. S.E., Minneapolis, MN 55455.

Sept 8 HAUCC will meet at 7:30 PM in Rm 117 of the Science & Re-

"Our goal was to produce 100% reliable business programs."



"What do we mean by reliable programs? Three things: good program design, documentation, and full support.

DESIGN Good program design meets a wide variety of customer needs without reprogramming.

Keith Parsons, President
Alan Cooper, VP, Systems Development

CIRCLE INQUIRY NO. 49

Our programs are comprehensive yet retain their flexibility. They allow convenient backup, are easy to use and have been thoroughly tested and field proven.

DOCUMENTATION We consider the quality of the documentation to be as important as the programs themselves. That's why our manuals are clear, concise and complete.

SUPPORT And when it comes to support we're second to none. We release periodic updates, answer your questions and are available to provide technical assistance. Now *that's* reliable."

Our growing Business Systems series currently includes: **GENERAL LEDGER, ACCOUNTS RECEIVABLE, NAD** (Name and Address File system), **QSORT** (full disk sort/merge), and **CBASIC** (a powerful business Basic). For details, contact our sales manager, Richard Ellman.

Structured Systems Group

5615 KALES AVE. DEPT. IA6 OAKLAND, CA 94618 (415) 547-1567

All systems are compatible with any Z-80 or 8080 CP/M™ system.

search Bldg. of the main campus of the Univ. of Houston. For more details write or call P.O. Box 37201, Houston, TX 77036, (713) 661-6806.

Sept 8 Northern New Jersey Amateur Computer Club (NNJACC) will hold its meeting at the Fairleigh Dickenson University, on the Rutherford Campus, Becton Hall, Room B8, at 7 P.M. For details write NNJACC, 593 New York Ave., Lyndhurst, NJ 07071.

Sept 9 The Permian Basin Computer Group — Odessa Chapter meets at 1 P.M. in the Electronic Technology Bldg., Room 203 on the Odessa College campus. For details contact John Rabenaldt, Box 3912, Odessa, TX 79760, (915) 332-9151.

Sept 10 North Orange County Computer Club will have its meeting at Chapman College, Orange, CA. Doors open at 12:00. 105 Hashinger Hall Auditorium. Membership Chairman, Tracey Lerocker, (714) 998-8080 evenings. For more information write P.O. Box 3603, Orange, CA 92655.

Sept 12 Electronic Conventions Inc. will meet through September 14 at Wescon/78 Show and Convention, Los Angeles Convention Center and Los Angeles Bonaventure Hotel. For more information contact William C. Weber, Jr., Gen. Mgr., Electronic Conventions, Inc., 999 N. Sepulveda Blvd., El Segundo, CA 90245.

Sept 12 Okaloosa Computer Hobbyist Club will meet in the Community Room of the First Federal Savings & Loan Assoc. of Okaloosa County, 158 Elgin Pkwy N.E., Ft. Walton Beach, FL at 7 P.M. For details call (904) 242-5938.

Sept 12 Rome Area Computer Enthusiasts (RACE) meets on the second Tuesday of every month at Patty's Stagecoach Inn at 7:30 P.M. For details contact Mike Troutman, RD 1, W. Carter Rd., Rome, NY 13440, (315) 336-0986.

Sept 13 Home Computers Users Group for Radio Shack TRS-80 meets at 7:30 PM. For details write or call TRS-80 Users Group Information of Eastern Massachusetts, c/o Miller, 61 Lake Shore Road, Natick, MA 01760, (617) 653-6136.

Sept 13 Homebrew Computer Club meeting will begin at 7 P.M. in Menlo Park, CA at the Stanford Linear Accelerator Center Auditorium. Contact the club at P.O. Box 626, Mountain View, CA 94042, (415) 967-6754 for details.

Sept 14 Mid America Computer Hobbyist meeting will be at 7:00 P.M. at Commercial Federal Savings & Loan, Bellevue NE. Intersection of Galvin Rd. and U.S. Hwy. 73-75. Write P.O. Box 13303, Omaha, NE 68113 for further information.

Sept 14 Utah Computer Association will meet at Murray High School, Rm 154, 5440 S. State St., Salt Lake City, UT at 7 P.M. For details write or call Larry or Holly Barney, 1928 S. 2600 E., Salt Lake City, UT 84108. (801) 485-3476.

Sept 14 The Rochester Area Microcomputer Society will meet at the RIT Campus, Rm. 1030, Bldg. 9 at 7:30 P.M. For details write RAMS, P.O. Box D, Rochester, NY 14609.

Sept 14 North Florida Computer Society will meet at 227 Edison Dr., Pensacola, FL 32505. For details write this address or call Eugene Rhodes at (904) 453-3844.

Sept 15 Long Island Computer Association meets at 7 PM at the New York Institute of Technology, Old Westbury Campus, Route 25A between Route 107 and Glen Cove Rd., Rm. 508. For more details write Long Island Computer Association, 36 Irene Lane East, Plainview, NY 11803.

Sept 15 Amateur Computer Group of New Jersey (ACGNJ) meets at UCTI, 1776 Raritan Rd., Scotch Plains, NJ 07076 at 7 P.M. For further information write to the club at the above address.

Sept 16 San Diego Computer Society will meet at the Grossmont Community College Student Center, 8800 Grossmont College Dr., El Cajon, CA. Doors open at 12:30. For details write P.O. Box 9988, San Diego, CA 92109, or call (714) 565-1738.

Sept 16 The 7C's Committee (Affiliated with the Cleveland Digital Group) will meet at Cleveland State University Student Services Bldg., in the Kiva Room at 2:00 P.M. For more information write to Cleveland Digital Group, 8700 Harvard Ave., Cleveland, OH 44105.

Sept 16 Philadelphia Area Computer Society will meet at 2 PM at LaSalle College Science Bldg. at the corner of 20th & Olney Ave. For more details write PACS, P.O. Box 1954, Philadelphia, PA 19105.

Sept 16 Computer Hobbyist Group of North Texas will meet at UTA University Hall, Rm 108 at 1 PM in Arlington, TX. For details contact Neil Ferguson at P.O. Box 1344, Grand Prairie, TX 75051, (817) 387-0612.

Sept 17 Central Florida Computer Club will meet at 2010 Fosgate Dr., Winter Park, FL 32789 2:00 PM. Contact Bill Kerns for details.

Sept 17 Cleveland Digital Group meets at 2 P.M. in the old railroad station at Safier's Inc., 8700 Harvard Ave., Cleveland, OH 44105.

Write the club at this address for more information.

Sept 19 Rhode Island Computer Hobbyists (RICH) meets the at the Knight Campus of Rhode Island Junior College in the Faculty Cafeteria at 7:30 P.M. For details contact Emilio Iannucillo, RICH, P.O. Box 559, Bristol, RI 02809, or call (401) 253-5450.

Sept 21 Madison Computer Society will meet at 7:30 P.M. at 2707 McDivitt Rd., Madison, WI 53713. Mike Shoh, president.

Sept 21 Sacramento Pet Workshop meets from 7-10 P.M. every third Thursday of the month. For more information contact David Howe, (916) 445-7926.

Sept 22 Alamo Computer Enthusiast meets at 7:30 PM in Rm 104 at Chapman Graduate Center at Trinity University, San Antonio, TX. For details call (512) 532-2340, or write to the club at 7517 Jonquill, San Antonio, TX 78233.

Sept 22 Washington Amateur Computer Society will meet at the Catholic University of America, St. Johns Hall, located at Michigan and Harewood Aves. in Washington, D.C. Contact Bill Stewart at (202) 722-0210 for club details between the hours of 10 A.M. and 12 P.M.

WHITE COLLAR MICROCOMPUTER

By James S. White

For the computer novice, reading manufacturer's literature is one of the better ways to learn about computing systems: what computers can and can't do and what kinds of computers are best for certain applications. Promotional literature tells which features and capabilities are available in today's market.

Reviewing the literature of several manufacturers can also provide information on what computers cannot do well, features and capabilities that are unavailable, that will be relatively untested or that will limit computer system vendor alternatives to very few. Finally, although an individual vendor's claims of superiority aren't necessarily true or relevant, these claims do raise questions a buyer might consider asking.

To keep the proper perspective, reading is probably a good way to learn about computing, just as is true of most other fields. Experience is usually a better teacher, as is talking to vendors and users. But reading is a good preparation for these activities.

A prime example of valuable free literature is a booklet available from DEC (Digital Equipment Corporation, with offices in major cities). "The Beginner's Guide to Small Business Computers" is a 14-page brochure written for managers of small businesses. It is designed to answer the questions, "How can a computer fit my particular operation? What exactly are the benefits to me? How do I go about getting one successfully installed?"

Several possible benefits of a small business computer are explained. DEC and others properly promote these listed benefits as criteria for computer justification. However, a prospective buyer can also well use such criteria to determine what types of characteristics his computer system should have, based on specifically identified potential uses.

More detailed computer application understanding is furnished by the brochure's section which discusses the matching of computer capabilities with the needs of a given business. These remarks are prefaced with the observation that an executive should not need to be concerned with details of how a computer operates. His emphasis should be on determining how a computer can best help in his business.

The final major section is titled "How to Acquire a Computer System", beginning with "Self Evaluation" and "Feasibility Study" topics. DEC proposes soliciting total system proposals from several vendors, a very good idea which also indicates the objectivity of this booklet. This section concludes with "Implementation" and "Internal Control" discussions, topics which are too often omitted from initial computer planning considerations.

Other DEC literature may also be of value to the novice interested in business microcomputing. As the world's largest manufacturer and marketer of mini/micro computers, DEC has a broad product line, and one suit-

CIRCLE INQUIRY NO. 30

able for many application areas. Because DEC's products are based on customer response rather than speculative offerings, its literature emphasizes products which computer users are finding of practical value. Many of these products are supported by literature which, in comparison to much of that available elsewhere, is relatively complete and easy to understand.

DEC's hardware Handbooks, such as the "PDP-11 Computer Family Products and Services" manual, are perhaps the most useful as instructional manuals because they give a fairly detailed explanation of basic computing principles. The software handbooks and spare parts and accessory catalogs also help explain the total product line of this manufacturer.

The most important literature for a computer system buyer is that available from prospective vendors. . .these will be computer retailers or "system houses" who buy hardware, combine it with software and sell a complete package.

Although the literature is good, and the "price is right", most of DEC's products aren't viable alternatives for the typical prospective microcomputer user. Many of the minicomputers that DEC sells are designed and priced for users several times as large as the under-ten-employee user for which microcomputers are most appropriate. However, some microcomputer prospects will find, as they gain computing understanding, that the additional capabilities offered by minicomputers more than pay for themselves through increased profits or other business gains.

LOCAL VENDORS

The most important literature for any prospective computer system buyer is that available from his prospective vendors. Generally, these will be either local computer retailers, or "system houses" who buy hardware, combine it with another vendor's or their own software, and sell a complete package. Locally available literature has two special advantages as a training tool: you can go to the source in case of questions on confusing or inadequately covered points, and the learning is very relevant to your forthcoming vendor selection.

In fact, the amount you are able to learn from a local vendor's literature and the related questions that you ask may properly be a prime vendor selection criteria. Because a good general understanding of computing concepts is very important to your successful use of a computer, the vendor and system that you can best understand may well be the best choice if its price and performance are satisfactory.

A final category of literature for prospective business microcomputer users deals with the books written specifically for this audience. Today there are books on large scale computing and microcomputing books generally oriented to the hobbyist. There are no introductory business microcomputing books generally available. However, by this fall there should be at least three books available. Several well-known and experienced authors have been working for many months and their results should be very worthwhile.☐

 CIRCLE INQUIRY NO. 39

THE MIND REVOLUTION

By Merl Miller

Last month we started a discussion of the Turing test. If you really want to understand Artificial Intelligence (AI), you should at least be familiar with this test. I'll repeat the basics, and then we'll consider some of the problems associated with it.

The test presents a human being with a terminal which he can use to converse with two unknown sources. One of these sources is a human being, the other is a machine. The operator tries to guess which source is responding. If the correct source cannot be determined at least 50% of the time, the machine is said to simulate human behavior. (For a more complete description, please see my July column.)

AI research could accomplish many things, and one possible goal could be to produce a complete description of a machine capable of passing the test, or to prove that no machine could pass it. However, when dealing in this field one should keep in mind two things: 1) It may be that more than one type of machine can pass the test; 2) A lot of things were, at one time, impossible and great scholars provided "proof" that they were. Therefore, computerists should not be limited in their thinking by a lack of imagination.

This all leads to the most interesting question of all. If a machine passes the test, it means that there is at least one machine capable of solving problems as well as a human being. Does this then mean that we can build a machine that is capable of solving problems humans can't?

The intellectual capabilities of a human being are directly related to the brain. The brain has a finite size, mass and number of cells arranged in a definite structure. We now have the capability of putting over 200,000 transistors, resistors and circuits on a single microprocessor chip, so we might be able to simulate the electronic movement of the brain. We might be able to "produce" intelligence in a computer, but the problem is we don't know if the electron movement is a function of intelligence or if it *is* the intelligence.

Machines have been built that can learn to produce solutions to specific intellectual problems which are superior to the solutions people produce. But, solving problems is a limited intellectual capability. What we need to do is to build a machine that can adjust or adapt to its environment as human beings essentially do.

Surprisingly little is known about the limitations of human intelligence. The ability to solve certain types of problems is regularly tested, but is this a true test of intelligence? The major shortcoming of this kind of test is that it doesn't measure what the person is capable of learning. Nor does it measure how the individual adapts

to his environment. What we need is a better description of how the mind works.

Perhaps the problem with intelligence is that it cannot be explained logically. If this is true, then there must be some types of knowledge that machines cannot possess, some methods of gaining knowledge that machines cannot employ. What is important is whether there are some forms of learning intelligence that can be exhibited by machines. Usually, if you are going to design a machine that does something, it is a good idea to try doing it yourself and seeing what happens. This does not mean your machine will wind up imitating a human approach; actually, machines often work more efficiently on some problems when they operate in ways that are unreasonable for human beings.

If you would like to start your own AI research, you might begin by examining and attempting to explain these four types of learning:

1. Subconscious learning, in which knowledge is somehow obtained without conscious reasoning.
2. Emotional learning, which is perceived as an emotion without reasoning.
3. Inspired learning, in which knowledge is given to you instantaneously.
4. Paradoxical learning, in which one is able to perceive knowledge that is self-contradictory, regardless of how it is expressed in words.

Is a machine capable of these types of learning? Is it capable of learning anything? Or is it only capable of doing what it is programmed? If you tell a three-year-old, "Do not go into the street because a truck will run over you; we don't want any flat little people around here," does he/she learn or is he/she programmed? If you tell a robot, "Do not go into the street because a truck will run over you; we don't want any flat robots around here," does it learn or is it programmed?

I don't have any answers — just questions. They are kind of interesting, though, aren't they? If you have some answers, please write me. Merl K. Miller, President, Matrix Publishers, inc., 30 N.W. 23rd Place, Portland, OR 97210.□

 CIRCLE INQUIRY NO. 50

COSMAC VIP

$249* gets the entire family into creating video games, graphics and control functions. For starters.

COSMAC VIP, the completely assembled, ready-to-operate RCA Video Interface Processor, opens up a whole new world of computer excitement. New challenges in graphics, games and control functions. Yet it's just $249.00.

Easy to buy. And easy to program, thanks to its unique, easy-to-use interpretive language. You get a complete how-to book including programs for 20 games: fun, challenging, and ready to load and record on your cassette.

Simple but powerful.

Built around an RCA COSMAC microprocessor, the VIP is a complete computer system that can grow with you. It has 2K of RAM, expandable on-board to 4K. Plus a ROM monitor, audio tone output to a built-in speaker, power supply, and 8-bit input and output ports for control of relays, sensors, or other peripherals.

Soon RCA will offer options for color graphics and 256 tone sound generation. An optional auxiliary keyboard will open up an exciting world of two-player games.

Take the first step now.

Check your local computer store or electronics distributor for the VIP. Or contact RCA VIP Marketing, New Holland Avenue, Lancaster, PA 17604. Phone (717) 291-5848.

*Suggested retail price. Does not include video monitor or cassette recorder.

The fun way into computers.

RCA

CIRCLE INQUIRY NO. 43

JURISPRUDENT COMPUTERIST

By Elliott MacLennan
Attorney-at-Law
Stephen Murtha

BUSINESS PLANS

A business plan in its simplest form is a statement of objectives; in a more sophisticated form, it is a detailed flowchart of how the business is going to succeed. A business plan is a valuable asset for any entrepreneur, regardless of how long he has been in business.

The business plan is of interest and service to two groups of people involved in business. The first is management. Used properly as a tool, it will enable the people running the business to periodically raise their heads above the day-to-day operations, take a look where the business is heading and make sure it is on track. Using foresight is one important characteristic which entrepreneurs often lack.

The second group of people to whom the business plan is of value are potential investors. A complete and well documented business plan can be a powerful sales tool. You are much more likely to get someone to invest in your business if you can show a detailed plan of operation, one that shows not only a thorough plan of the operations but also covers all possible contingencies. The moment a potential investor asks "What would happen in this situation?" and your reply is "I don't know.", you are in trouble.

The importance of a business plan cannot be underemphasized. Entrepreneurs usually avoid making business plans for two reasons. The first is that they think the process is too involved and they don't feel that the payoff is going to be worth the effort. Therefore, a business plan should only include as much data as is required for the given circumstances.

The second barrier to the entrepreneur is the uncertainty which surrounds the future. Fortunately, the business plan has an element of self-fulfilling prophecy about it. The more often it is consulted and the variances explored to find their source, the closer reality will look to the business plan.

There are a number of items which should be covered in any complete business plan. The following is a list of the main topics. Please note that it is not all inclusive and certain applications will require additional information. It is, however, a good list of basic items.

- The products and/or services
- The markets to be served
- Marketing strategy
- Staff and plant requirements
- R & D programs and future expansion
- Pro forma financial statements
- Capital needs and sources
- Legal structure of the business
- The professional advisors and outside help

The products and/or services: For more computer companies this is where the business begins. It is the birth of a new piece of hardware, a more complete or efficient software package, or a needed service which motivates the entrepreneur into establishing a business. This part of the business plan should include the specifics of the particular product or service and define them as clearly as possible. Items such as price, quality, customer service, compatibility with other related products on the market, etc. should be thoroughly spelled out. The state-of-the-art should be compared to the product or service to determine how long a life-span can be expected. All specifications, prices, prototypes, etc. should be as detailed as possible so that other factors such as marketing strategy or production requirements can be developed. In short, as clear a picture of the product as possible should be drawn at this point so that the rest of the plan has some validity.

The markets to be served: Determining what market the business will concentrate on is the first step in developing a viable marketing program. The plan should specify the market which the business will focus on. The potential rewards, pitfalls, and characteristics of each of the markets should be spelled out. Major competitors, potential customers, and suppliers should be listed in as much detail as possible.

Marketing strategy: Marketing strategy is the game plan which you intend to follow in order to get your chosen market to purchase your products or services. A complete marketing strategy should include topics such as promotional methods, advertising plans, methods of product distribution, and terms of sale. The plans should be as specific as possible with such details as the media advertising that will be used, whether products should be sold or leased, whether sales will be directed toward customers or whether some intermediary must be used.

The personnel required to do a good marketing job should also be considered. Sometimes a salaried or commission salesman can be retained by the business if it is large enough. If not, contacting buyers directly via direct mail or selling indirectly through retail outlets may be the only alternative.

Staff and plant requirements: This section should include the area of responsibility of each person on the management team. Not only should each person be assigned a specific area of responsibility, but all tasks should be delegated to someone on the team or provisions should be made for their completion outside of the organization. An organization chart is generally of service to clarify the relationships between the team members.

The plant requirements should be considered in light of the amount and kind of space required for the business. For many new manufacturing firms, the owner's garage may be acceptable at first. A retail store, on the other hand, should have as good a location as possible. Consideration should be given to future expansion when choosing a site and determining the details such as length of the lease and amount of space.

There are many factors dealing with the successful business plan. We will continue looking at them in the next column.□

The material presented in this column is intended for the reader's general information. The authors request that the reader consult professional advisors prior to applying this material to his or her specific situation. Anyone needing further information can contact the authors directly at:

Elliott MacLennan	Stephen Murtha
2855 Mitchell Dr.	3 Altarinda Dr.
Suite 130	Suite 304
Walnut Creek, CA 94598	Orinda, CA 94563

FROM THE FOUNTAINHEAD

By Adam Osborne

I have frequently criticized the small, new manufacturers of microcomputer hardware for their less than professional business practices. Some of the "old line" hardware manufacturers now selling to the microcomputer industry are less than fair. It has come to my attention that many of the established hardware manufacturers start their **warranty period** the day that they ship hardware to a computer store. Many of these warranty periods are only three months long. If a computer store keeps the hardware in inventory for three months, you, the customer, buy a piece of brand new hardware with no warranty.

So once again, buyers beware. When you buy hardware that is supposed to have a warranty, make certain you know when the warranty period began and when it is going to expire. Computer stores, in turn, should get together and force vendors to measure hardware warranty periods from the day the product is sold to an end user. Hardware manufacturers have no grounds for claiming that they cannot discriminate between an end user as against a customer buying for retail. Retail customers buy for resale, end users do not.

Technical Design Labs is one of the companies about whom I have had frequent complaints regarding products that are not delivered, or do not work when delivered. It was in financial trouble and had operating problems which combined to cause its difficulties. However, it has been acquired by Xitan, Inc., a new company with adequate cash and management experience to solve TDL's problems. I am hopeful that it will once again join the ranks of manufacturers with a happy customer base. Technical Design Labs has a good product; it simply did not have the cash to operate in a sound fiscal fashion.

One of the most interesting phenomena of the microcomputer industry seems to be the inherent **reliability of hardware**. Considering that most hardware was manufactured by amateurs, or semi-professionals at best, I would have expected formidable service problems to follow installation of microcomputer systems. This does not seem to be happening. People buying minicomputer systems and mainframe computer systems seem to be far more concerned with service problems than microcomputer customers. Perhaps this is because microprocessors, and LSI devices in general, are so reliable. Intel claims that based on the rate of LSI device failures they have experienced thus far, 90% of all microprocessor and support devices that they have shipped to date will still be working in 500 years.

It appears as though LSI devices either do not work initially, or else they work forever. Therefore, when you get a new microcomputer system, you can expect trouble for the first few months while marginal parts fail, but after this burn-in period, you can expect your electronics to be extremely reliable.

The microcomputer industry is still too new for anyone to be sure that this diagnosis of reliability or failure is in fact accurate. But if it is, it will have far reaching effects both on hardware service practices and on the design of hardware itself.

Minicomputer and mainframe computer manufacturers operate service departments that expect a certain number of electronic failures to occur every year in every piece of hardware sold. Most microcomputer manufacturers never bothered with the hardware service problem, not because they understood that there would be no problem, but rather because they did not know how to cope with it. Rather than setting up elaborate anticipatory service organizations, they waited for the problem to arise, figuring that they would solve it when it happened. Meanwhile, they addressed more pressing problems. But the problem never happened. Considering the amateurish design that characterizes so much of the microcomputer hardware on the market today, it is remarkable how reliable the hardware is.

I have a theory to account for this reliability. One of the consistently amateurish characteristics of the logic design that we see in microcomputer hardware is utter simplicity.

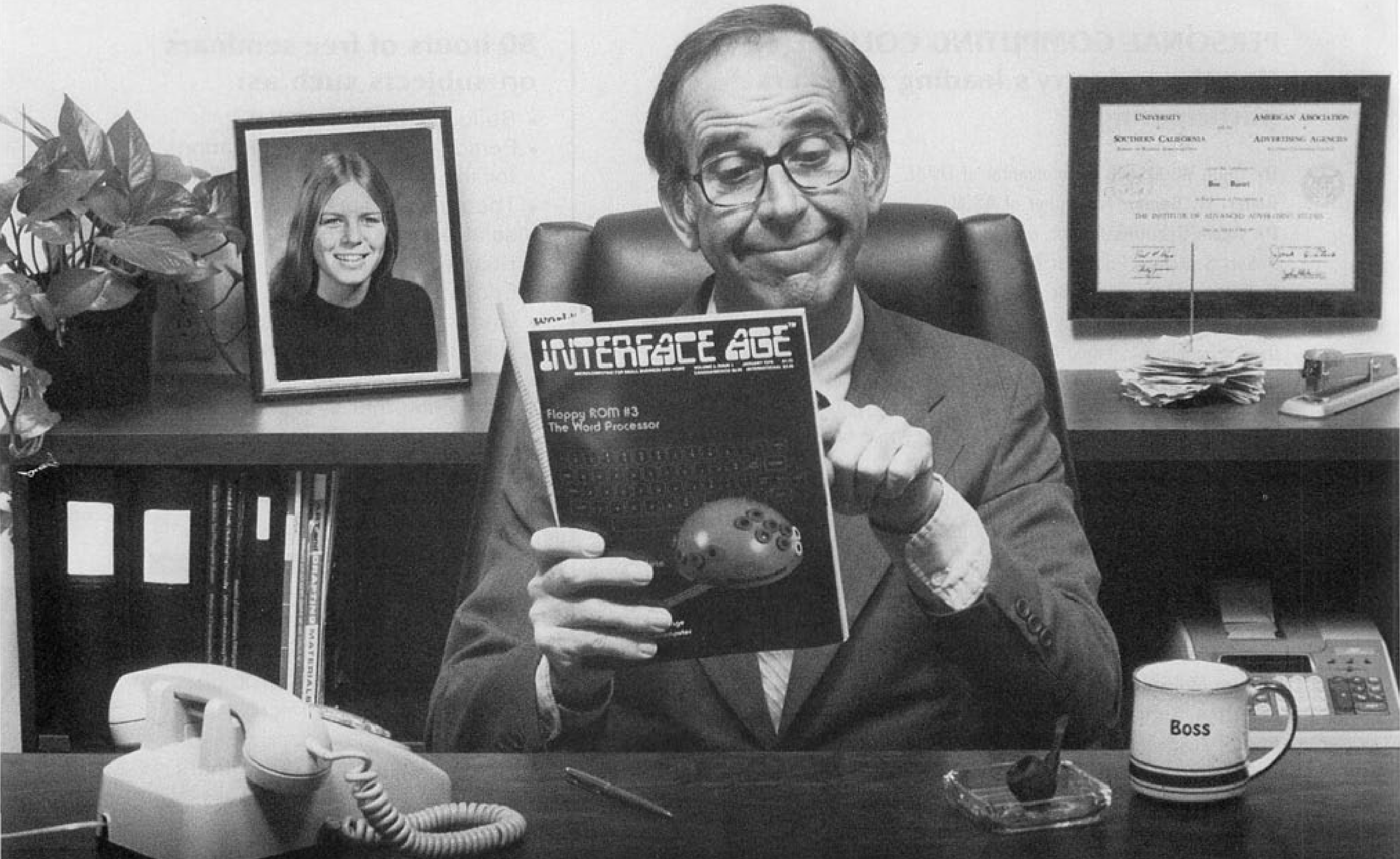
Consider information transferred across a parallel bus. Your microcomputer system, in all probability, assumes that the data it transmits will be received and that nothing will happen to the data in transit. But mainframe manufacturers make no such assumptions. They check parity, and probably have special logic that attempts to replace a missing bit.

If a bit goes bad in a memory chip on a $300 microcomputer memory board, you may as well shut down until you replace the bad chip. But any mainframe memory modules will have special logic that flags the existence of a bad bit and subs for it until you get around to replacing the bad memory chip with a good one.

I believe that the amateur microcomputer designers' "error" of simplicity was no error at all. Rather, they lucked out, and without knowing what they were doing, they did the right thing. LSI components do

INTERFACE AGE

The magazine that puts microcomputing in your hands.



Technology has brought the computer within your grasp. **Interface Age** puts it in your hands.

The small businessman like yourself is aware of the progress made in microcomputing.

What you may not have is an update of what is happening in microcomputer development, and what discoveries might save you money and increase your computer's usefulness.

Interface Age has the information you need.

In issues you've missed we've presented...

- A way to improve your chances of success in a business venture.
- A complete general ledger and payroll accounting program with documentation.
- A new design for typewriter keyboards that obsoletes the electric typewriter.
- A program that can assist you in writing reports and letters.
- A program to plan your investment in income property.

Articles ranging from the fundamentals of computers to languages and system design, tutorials, activities, and new product releases to help you get started in microcomputing and keep up with the industry.

Get a hold of the information you need. Subscribe to **Interface Age** today. Save $7 over newsstand price.
12 Monthly Issues: $14 U.S.; $16 Canada/Mexico; $24 International. 24 Issues: $24 U.S.; $28 Canada/Mexico. 36 Issues: $36 U.S.; $42 Canada/Mexico.

Name____________________
Address____________________
City__________ State______ Zip______
☐ Visa Card ☐ Master Charge
Acct. No. ______________ Exp. Date______
Signature____________________
Check or money order (U.S. Funds drawn on U.S. Bank)
Make checks payable to: INTERFACE AGE MAGAZINE
P.O. Box 1234 Dept. IA8 Cerritos, CA 90701
☐ *Please send information on issues available back to 1975.*

seem to be inherently reliable devices. Moreover, the probability of an error in a piece of electronics is a function of the number of pin-to-pin connections. Microprocessor-based electronics is very simple, and has few pin-to-pin connections. In consequence, extra error checking logic introduces more problems than it resolves.

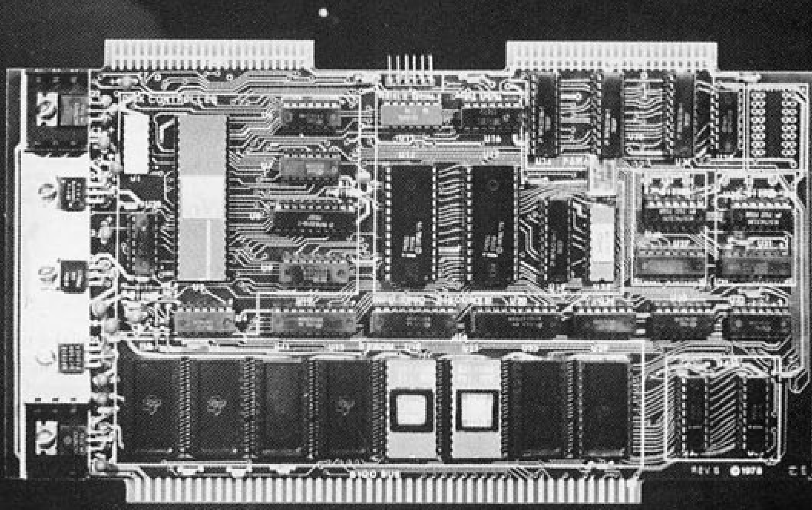
The probability of an error in a piece of electronics is a function of the number of pin-to-pin connections.

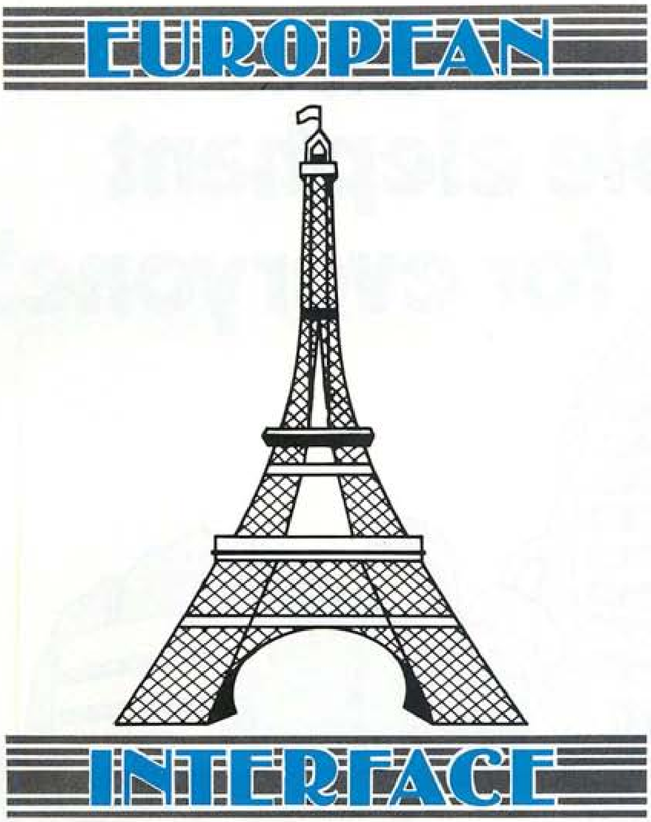
Of course, a number of microcomputer manufacturers simply goofed. For example, one well-known manufacturer was selling a board that occasionally allowed 60 volts to reach a ROM chip. The ROM chip quickly failed. Strange to say, this particular manufacturer always had a good supply of replacement ROM chips, one of the few parts that you could order and expect prompt delivery.

Jim Warren, who runs the West Coast Computer Faires and has been editor of Dr. Dobb's Journal, is planning to start the **first microcomputer user's newspaper**. His newspaper, which will run bi-weekly, will differ from microcomputer magazines by providing topical news items rather than technical articles and feature columns. Jim is one person who has made a really significant contribution to the microcomputer industry and I expect that this newspaper will do very well. Jim may be reached at:

345 Swett Road
Woodside CA 94062

□





By Hans Drewitz and Roland Hesse

Recently a group of journalists from Zero-Un-Informatique, a well-known French data processing magazine, invited some of the active people in the field of microcomputers to share ideas on the industry in France. Although most of us had met before (the personal computing community in France is still small), it was the first significant organized get-together and the atmosphere and enthusiasm reminded me of the pioneer days in the personal computing field of the U.S.A. There was a feeling of openness, a willingness to help others in the resolution of problems and, above all, a common interest in creating a momentum in this discipline.

We discussed the problems which so far have slowed down the popularity of microcomputers. Some of these problems, such as price and language, have been discussed in previous articles of INTERFACE AGE. Other topics dealt with consumer knowledge. One problem is the wide variety of products the inexperienced European end user or distributor has to choose from.

In America, even with the enormous speed with which this field has developed, it was still an evolutionary process which lead people into personal computing and educated the public at the same time. It all started with 8080s from ALTAIR and IMSAI and led to the ready-made home computer like PET on one hand, and the sophisticated timesharing system such as the ALPHA MICRO SYSTEM on the other. All this happened within two years and the people had some time to evaluate these products.

In Europe today, a whole variety of equipment hits the public at the same time, and the process of selection and consumer education is much, much slower. There is a big job to be done by clubs, computer shops and journalists in assisting the public to resolve this problem. In this sense the meeting organized by Zero-Un-Informatique was very helpful and should be repeated.□

DESCRIPTION OF TOUR PACKAGES

MICRO PHILADELPHIA

Includes: Five (5) nights accommodations

Choice of ONE:

a) 2½-hour tour of historic Philadelphia, including Independence Hall, Liberty Bell and other Colonial landmarks (daily, year-round)
b) 2½-hour tour of modern Philadelphia, including Boat House Row on the Schuylkill, Philadelphia Museum of Art, the new Mint and several of the city's new buildings and plazas (daily, year-round)

COST: From $108.00/person sharing a room, add'l night @ $20.00/person
From $168.00/person single room, add'l night @ $32.00/person

MINI PHILADELPHIA

Includes: Five (5) nights accommodations

Choice of ONE:

a) 7-hour tour of old Philadelphia and Valley Forge (daily, April 17-October 30)
b) 7-hour tour of modern Philadelphia and Valley Forge (daily, April 17-October 30)
c) Dinner at Old Original Bookbinders, one of the city's landmarks; includes soup, seafood entree, dessert, tip and tax (nightly)
d) Penthouse dinner, with spectacular skyline view includes specialty appetizer, choice of entree and dessert (excluding flambes), tip and tax (nightly)
e) Dinner at Spats; includes choice of appetizer, entree and dessert, tax and tip (nightly except Sunday)
f) Dinner at charming, medieval Monk's Inn; includes one drink, choice of appetizer, salad, entree, vegetable, beverage, dessert, one glass of wine, tip and tax (nightly)

Choice of TWO:

a) 5-hour motorcoach tour of historic and modern Philadelphia (daily, year-round)
b) Half-day tour to George Washington's encampment at Valley Forge and surrounding area (daily, April 17-October 30)
c) Full day at Great Adventure entertainment park, includes admission to park and African safari (optional round-trip bus fare from Philadelphia not included); open 10 a.m. to 10 p.m., daily May-August, weekends only late April, September and October.
d) Lunch at Spats Restaurant; includes choice of appetizer, entree and dessert, tip and tax (daily except Sunday).
e) Dinner at Middle East restaurant; includes entertainment, choice of dinner menu, tip and tax (nightly)

Choice of ONE:

a) 2½-hour tour of historic Philadelphia (daily, year-round)
b) 2½-hour tour of modern Philadelphia (daily, year-round)
c) Box seat at a Phillies regular season home game (April-September only, must be reserved at time of booking, after home game schedule has been consulted)

COST: From $153.00/person sharing a room, add'l night @ $20.00/person
From $213.00/person single room, add'l night @ $32.00/person

MACRO I PHILADELPHIA AND NEW YORK

Includes: Philadelphia — same features as Mini Philadelphia.
New York includes:

2 nights at the Americana City Squire Inn

Orchestra seat at an evening performance of a Broadway musical of your choice*

Orchestra seat at an evening performance of a Broadway comedy or drama of your choice*

After-theater snack at Gallagher's Restaurant, from 10:30 p.m., includes tip and tax

United Nations tour

All hotel taxes

*Broadway theater reservations should be made in advance. Because of constant changes in Broadway theater prices, a small surcharge may be necessary, in which case you will be advised at the time of confirmation.

COST: From $246.25/person sharing a room 7 nights accommodations, additional night in Philadelphia is $20.00/person and in New York additional night @ $30.75/person

From $336.25/person sharing a room 7 nights accommodations, additional night in Philadelphia is $32.00/person and in New York additional night @ $51.80/person

MACRO II PHILADELPHIA, BOSTON AND NEW ENGLAND

Includes: Philadelphia — same features as Mini Philadelphia. Boston and New England include:

3 nights' first class accommodations at hotel selected

Round-trip transfer from Logan Airport to downtown Boston

Boston Traveler's Information Kit

Dining certificate worth $10.00 toward a meal at Anthony's Pier 4

Choice of THREE:

a) 3-hour tour of greater Boston and Cambridge (daily, May 15-October 31; departs on the hour, 9 a.m. through 2 p.m.; Boston Tea Party Ship admission not included)
b) 3-hour tour of Lexington and Concord (daily, 1:30 p.m., May 15-October 31; admissions not included)
c) 7-hour tour of Boston, Cambridge, Lexington and Concord (May 15-October 31, daily, 9 a.m.; admission to Boston Tea Party Ship and luncheon not included)
d) 4-hour tour to Quincy and Plymouth (May 15-October 18; Tuesday, Thursday, Sunday, 12:45 p.m.; admission to Quincy Mansion and Mayflower II not included)
e) 8½-hour tour to Cape Cod and Hyannisport (May 15-October 31, Monday, Wednesday, Thursday, Friday, Saturday, 8:30 a.m., admissions and luncheon not included)
f) 4-hour tour to Salem and Marblehead (May 15-October 15, Monday, Wednesday, Friday, Saturday, 12:30 p.m.; admission to Witch House not included)
g) 10-hour tour to Martha's Vineyard (May 15-September 3, Tuesday through Saturday, 8:15 a.m., ferry boat fare included, luncheon not included)
h) 6½-hour Rockport, Gloucester and Cape Ann evening tour (May 22-September 3, daily, 4:30 p.m.; September 4-October 14, daily, 3:30 p.m.; dinner not included)
i) 8-hour tour along coast of Massachusetts, New Hampshire and Maine (May 15-September 24, Monday, Tuesday, Wednesday, Friday, Sunday, 9 a.m.; September 25-October 15, daily, 9 a.m.; luncheon not included)
j) 8-hour tour to Newport, R.I. to see Gatsby-era mansions (June 11-September 3, Thursday & Sunday, 9 a.m.; admission to Vanderbilt's Marble House and luncheon not included)
k) 10-hour New England tour to see the fall foliage (September 13-October 16, daily, 8:30 a.m.; luncheon not included)
l) 8-hour tour to Old Sturbridge Village, a museum village of the early 19th century (May 15-October 22, daily, 10:30 a.m.; admission to village included; luncheon not included)

Choice of ONE Boston area admission:

a) Boston Tea Party Ship and Museum
b) Museum of Science
c) Museum of Fine Arts
d) Plymouth National Wax Museum
e) Mayflower II replica of original Mayflower
f) Plimoth Plantation
g) Salem Witch Museum
h) John Hancock Observatory
i) Institute of Contemporary Art
j) "The Whites of Their Eyes" Pavilion
k) "Where's Boston?" Pavilion
l) Prudential 52nd Floor Skywalk
m) "USS Constitution" Museum
n) Museum of Transportation
o) Children's Museum

Choice of ONE:

a) Dinner and dancing at Top of the Hub Restaurant (must be confirmed at time of booking)
b) $10 voucher, good toward orchestra tickets to pre-Broadway show, and regular tickets to Summer Theater performances, to the Boston Pops or to Red Sox home games (seasonal options for which dates must be checked and reservations made at time of booking)

Hotel tax

COST: From $302/person sharing a room, additional night in Philadelphia is $20.00/person and additional night in Boston is $25.00/person

From $404/person single room, additional night in Philadelphia is $32.00/person and additional night in Boston is $40.00/person

Three (3) persons sharing a room is even lower.

The Amazing Micro Mouse Maze Contest

IEEE Spectrum/Computer magazine's Amazing Micro Mouse Maze Contest, the brainchild of Spectrum's Editor Donald Christiansen, was spawned about a year ago as a challenge to engineering persons to design a self-contained maze-solving electronic mouse, within prescribed contest rules. The fastest mouse through the maze wins the $1,000 Grand Prize. Roger Allan, Spectrum's Associate Editor, has been managing the contest nearly since its beginning, handling all of its apsects.

Impetus for the contest was provided by an earlier mechanical-mouse contest by Machine Design magazine. In that one, which was much simpler than ours, a winning mechanical mouse would race down a runway of known length and composition, climb a pole at the runway's end, go up to a clock, and strike the clock's bell, in the fastest time.

Although the Amazing Micro Mouse Maze Contest was conceived about a year ago, it did not take off until early January 1978, in response to an advertisement placed in the January 1978 Spectrum. By the closing date of March 31, 1978, nearly 6,000 entrants were registered worldwide, nearly half of whom were not IEEE members. Of the total, about 300 were overseas contestants. Contestants paid $3.95 and received the contest rules plus a starter kit of parts (taken from an assortment of microprocessors, RAMs, PROMs, I/O ports, bus transceivers, batteries, etc.) provided through us by leading components manufacturers.

Maze specifications are as follows: Overall dimensions are not to exceed 20 by 20 feet. The maze cross section (see Figure 1) consists of a running surface of black, #00 sandpaper, nominally level. Side walls are white and the top of the walls are red. Temperature and illumination are room ambient. More than one path to maze exit may be possible, with no deliberate orientation with respect to magnetic north. The maze will contain these elements: Elbow, tee, U, "mouse-trap," and straightaway (see Figure 2). An 8-inch high guidepost (red) approximately 1-inch square will be placed 24 inches beyond the finish line at the center of the finish-line channel.

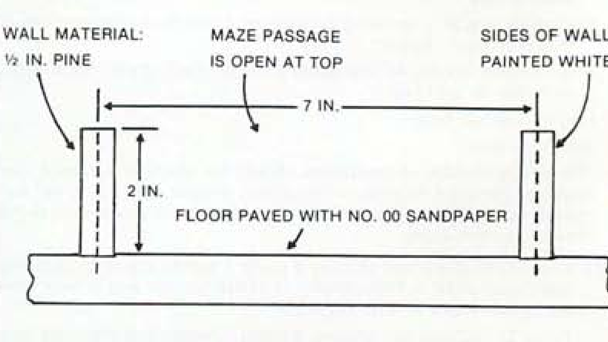


Figure 1

"A splendid performance in three acts"

ACT-I

Known for its dependability, ease of interfacing, utility and affordable price, the ACT-I enjoys its reputation as one of the most popular "glass teletypes" on the market. If your computer system communicates in serial ASCII, the ACT-I could be just the tool you need to get online.

The ACT-I computer terminal manages a 1024 character display organized as 16 lines of 64 characters selected from the standard upper case ASCII set. Receipt of more than 64 characters on a line or the Line Feed code initiates a scroll operation.

STANDARD ACT-I FEATURES INCLUDE:

Switch selectable data rates of: 110, 300, 600, 1200, 2400, 4800, 9600, and 19200 Baud.

Switch selectable UART options: Odd, even, or no parity, one or two stop bits.

Jumper Selectable Interface: RS232C, 20MA current loop or TTL voltage levels.

- Handsome, rugged, lightweight aluminum cabinet
- Standalone operation — absolutely no processor overhead required
- Highly reliable keyboard with two key rollover
- Clear sharp video output signal (RS170 standard) capable of driving any CRT monitor

Price $400. A cursor control / bell option is available for $25.00.

ACT-II

We've added the convenience of an acoustically coupled modem to the economy and performance of the ACT-I to create the ACT-II. Designed to communicate either with remote processors through its modem, or with local computers via its RS232C or 20MA current-loop interfaces, the ACT-II offers versatility unheard of at its low price. The ACT-II (without monitor) slips easily into an attache case (4 × 14 × 11 inches) to commute with you between work and home.

The ACT-II's demodulator employs four stages of active filtering to minimize the bit error rate of the receiver. If you are eager to join the ranks of those who sit at home and enjoy the use of a powerful computer system across town, the ACT-II can be your "password".

As a further convenience feature, the modulator input and demodulator output are available at jacks on the rear of the ACT-II cabinet so that you may link a local serial device (such as a digital casette tape or even your own computer system) to the remote computer through the internal modem.

The ACT-II can be purchased for only $550.00

ACT-IV

If you're looking for a low priced high powered terminal, consider these features which are all standard with MICRO-TERM's ACT-IV:

DISPLAY: Upper and descending lower case characters, 24 lines of 80 characters, and auto-scrolling.

KEYBOARD: Full ASCII with cursor controls and auto-repeat on several keys.

TRANSMISSION MODES: Character by character or "page" mode.

SPECIAL FUNCTIONS: relative and absolute cursor addressing, home up, erase to end of line, erase to end of screen, fixed tabs, report cursor position, and display control characters.

EDITING: in PAGE mode, the user can insert or delete characters on any line and insert or delete lines on the page.

DATA RATE: 300 to 19200 baud (Switch selectable on rear)

The ACT-IVa comes in a compact (briefcase compatible) cabinet without video monitor for $550.

The ACT-IVb comes complete with a 12" monitor and numeric keypad in a single enclosure for $800.

Optional available features: separate printer port (110-9600 baud) $50.

GENERAL INFORMATION:

All MICRO-TERM products are fully assembled, tested and guaranteed for 90 days. The entire MICRO-TERM product line is available from stock at discriminating computer stores or may be purchased directly from the factory. All prices are less monitors (which start at $130.00) F.O.B. St. Louis, Missouri.

VISA and Master Charge Accepted

MICRO-TERM INC.
P.O. BOX 9387
ST. LOUIS, MO 63117
(314) 645-3656

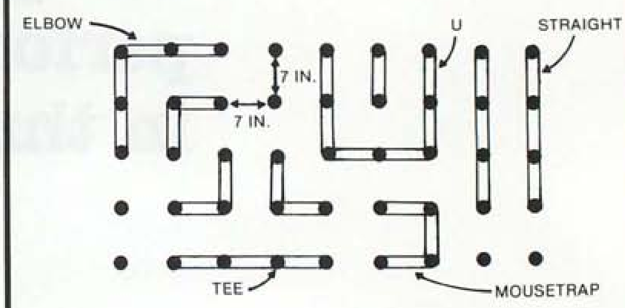


Figure 2

The mouse specifications are: width (superstructure) is 10 inches maximum; length (superstructure) also 10 inches maximum; height, no limit (a mouse that tips due to instability is disqualified). Motive power may be either batteries, electric motors or mechanical springs. No internal combustion engines are permitted. The mouse has to be completely self-contained with no hard wiring or radio communication to and from the mouse. Mazed configuration sensing is at the discretion of the builder; however, only inside (white) walls are employed as a sensing guide for physical sensing mechanisms. Optical sensors, on the other hand, are allowed to "peek" over the walls.

Procedures are as follows. The mice must be accepted and caged by the contest officials before the maze is unveiled and the runoffs begin. Handlers will place mice on a starting strip at the instruction of contest officials. Automatic timing mechanisms will register start and finish times. The fastest mouse first time through the maze wins the $1,000 grand prize. Another significant prize will be for the best learning mouse — the mouse who registers the best time in the last of three consecutive runs. (Additional special prizes will be awarded, including one for the most ingenious design.)

Mice may not be reclaimed, removed, or reprogrammed by owners and handlers between the three runs. Repairs and battery replacement will be permitted between runs but only under supervision of contest officials. A maximum 5-minute time limit is permitted on each run.

A series of trial runs are being planned, leading up to a final race. The trial runs will take place at major electronic conventions, around the country, at different times. All contestants are allowed to participate in any or all trial runs, leading up to the finals when the $1,000 Grand Prize will be awarded, and will be given the benefit of the fastest time achieved through the maze, at any of the trial runs. The maze is different for every trial run (it has reconfigurable walls for different maze configurations), and is kept under cover at the beginning of each trial run, until all trial run participants lock up their mice.

The first of these trial runs was held June 6-8, 1978, at the National Computer Conference's Personal Computing Section, in the Disneyland Hotel, Anaheim, California. Only six official mice were entered (out of a total of 54 that had been planned for up until the last few days before the initial trial run). Nearly all those 48 contestants that didn't quite make the run could not do so due to a lack of time to finish their designs. Nearly all of them are expected to make subsequent trial runs. In addition, a "show of hands" letter sent to about 50 percent

of all contestants around early March 1978 indicates that at least 350 to 400 contestants expect to have their mice designed and ready for entry by late summer to early winter 1978. Thus, it is expected that with each subsequent trial run, the number of entries will snowball into the finals, tentatively planned for the June 1979 National Computer Conference in New York City.

At the initial trial runs in Anaheim, California, one contestant's mouse made it through the maze in 51.4 seconds, while another's entry made it in 4 minutes and 32.48 seconds. Television coverage (KNX-TV, Los Angeles, CBS Channel 2) was ascertained. The TV coverage was run on an evening news program.

Beyond the second planned trial run at Personal Computing '78, Philadelphia, Pennsylvania, August 25-27, 1978, tentative plans call for trial runs at: Wescon, Los Angeles, California, September 12-14, 1978; the West Coast Computer Fair, Los Angeles, California, November 3-5, 1978; and Midcon, Dallas, Texas, December 12-14, 1978.

The enthusiasm this contest has generated has been enormous, not only amongst the contestants, but also amongst industry. For example, industry has been extremely helpful in providing us with a host of free material prizes awarded at the initial trial run; with electronic equipment to measure mice race-times and maintain contestant data files; with technical personnel to design and operate the race-time and contestant data systems; and of course with free electronic components, provided in the starter kits that were mailed to all contestants.

Spectrum will probably start another similar contest during 1979, as early indications show that a large number of persons would like to enroll in one, having missed the March 31, 1978 deadline for the Amazing Micro Mouse Maze Contest.□

EUROPEAN ROULETTE IN COLOR

By W. C. Hoffer

Fortunes have been made and lost at the roulette tables of the world, but after an initial investment for the hardware, you can play roulette without fear of losing your fortune. The game, European Roulette in Color, runs on the Compucolor microcomputer and requires a minimum of 16K bytes of user random access memory.

The original program was written in Dartmouth BASIC. The conversion to a non-pictorial version for Compucolor BASIC was not difficult. The time-consuming effort was animating the game and adding color. Currently, the "wheel" consists of a ball which rolls counter-clockwise on one circle, then clockwise on a smaller circle simulating how the ball falls into the winning number.

I have dispersed REMARK statements throughout the program to help the reader determine what happens in each section.

Operating instructions and playing rules are available at the beginning of each game. New players are urged to read them in order to avoid confusion.

However, the program is completely self-instructing and prompts the player for each input as required. Player inputs are checked for validity. Invalid plays are politely refused and the player is asked to play again.

After the playing surface appears on the screen, you will be prompted to PLACE YOUR BETS. The cursor will be positioned where a question is being asked, and the player must respond with either a YES or a NO each time the cursor points to a word. After YES answers are given to the ODD or EVEN, RED or BLACK display, a "$" will appear asking for the bet in dollars. A YES to the COLUMN question will result in a "1-2-3?" display, asking for the column of your choice and your bet. A YES to the NUMBER question will prompt "0-36?", asking for the number and your bet.

When all the bets have been placed, the ball will begin to roll just above the playing surface. When the ball stops, the winning number will appear on the left of the screen, and the BETTING RESULTS sign on the right. The actual results for each of your bets will follow that. Losses are displayed in RED and winnings in GREEN. The cumulative total for the games is kept for you and is constantly displayed on the right of the screen.

The HOUSE wishes you the best of luck and reminds you that you may pick up your winnings at the same location you deposited your losses.□

```
100 REM EUROPEAN ROULETTE
110 REM CONVERTED FROM DARTMOUTH BASIC BY:
115 REM W.C.HOFFER-2721 N. WANDA-SIMI VALLEY,CA-93065
116 REM REQUIRES 16K OF USER RAM
118 REM DISPLAY THE INTRODUCTION
120 PLOT6:PLOT33:PLOT12:PLOT27:PLOT11:PLOT14
130 PLOT3:PLOT21:PLOT6
140 PLOT6:PLOT25
150 PRINT"COMPUCOLOR PRESENTS EUROPEAN ROULETTE"
160 PLOT3:PLOT80:PLOT7
170 GOSUB 10000
190 PLOT6:PLOT7:PLOT12
210 X=0
220 PRINT "WELCOME TO THE COMPUCOLOR CASINO AND OUR EUROPEAN ROULETTE TABLE."
230 PRINT "WE WISH YOU THE BEST OF LUCK!."
240 PRINT:PRINT:PRINT
260 PRINT"DO YOU WANT INSTRUCTIONS";
265 PRINT
270 INPUT Z$
280 IF Z$="NO" THEN 460
290 IF Z$="YES" THEN 320
300 GOSUB 1960
310 GOTO 240
320 PLOT6:PLOT7:PLOT12
325 PRINT"    THIS IS A GAME OF ROULETTE.  YOU ARE ALLOWED TO BET ON"
330 PRINT"ANY (OR ALL) OF THE FOLLOWING: WHETHER A NUMBER IS ODD OR EVEN,"
340 PRINT"COLOR (RED OR BLACK) OF THE NUMBER, A COLUMN OF NUMBERS,"
350 PRINT"A NUMBER ITSELF.  NUMBERS RANGE FROM 0 TO 36.  IF A 0 APPEARS,"
360 PRINT"THE BANK COLLECTS ALL BETS EXCEPT THOSE BET ON THE NUMBER 0."
362 PRINT:PRINT
365 PRINT"THE PAYOFFS ARE AS FOLLOWS:"
370 PRINT"ODD OR EVEN   1 TO 1"
380 PRINT"RED OR BLACK  1 TO 1"
390 PRINT"A COLUMN      2 TO 1"
400 PRINT"A NUMBER     35 TO 1"
405 PRINT
410 PRINT"    YOU ARE ALLOWED TO BET FROM $1 TO $10,000. BUT THE"
420 PRINT"TABLE WILL ONLY ACCEPT BETS OF WHOLE DOLLARS."
425 PRINT
430 PRINT"(IF YOU WANT TO BET CHANGE--GO USE THE SLOT MACHINES)"
460 PRINT "HIT THE SPACE BAR WHEN YOU ARE READY"
470 R1=RND(1):IF INP(1)<>96 THEN 470
490 GOSUB 5000 :REM DRAW THE TABLE
500 REM TAKE THE BETS
501 PLOT3:PLOT0:PLOT10:PLOT6:PLOT7
502 PRINT"                    "
503 GOSUB 6300
504 GOSUB 6500
505 PLOT14
510 PLOT6:PLOT79
520 PLOT3:PLOT0:PLOT20
530 PRINT"PLACE YOUR BETS"
540 FOR J1=1 TO 500:NEXT J1
550 PLOT6:PLOT2
560 PLOT3:PLOT0:PLOT20
570 PRINT"PLACE YOUR BETS"
580 REM ODD?
590 PLOT6:PLOT1
600 PLOT3:PLOT4:PLOT24
610 INPUT"-";A$
620 IF A$="NO" THEN 760
630 IF A$="YES" THEN 660
640 GOSUB 1960
642 PLOT3:PLOT4:PLOT24
644 PRINT"        ";
650 GOTO 600
660 REM GET AMOUNT
665 B$="ODD"
670 PLOT3:PLOT9:PLOT24
680 INPUT"$";H
690 IF H<=10000 THEN 720
700 GOSUB 1980
702 PLOT3:PLOT9:PLOT24
704 PRINT"        ";
710 GOTO 670
720 IF H<0 THEN 740
730 IF H=INT(H) THEN 950
740 GOSUB 2010
750 GOTO 702
760 REM EVEN?
770 PLOT3:PLOT5:PLOT28
780 INPUT"-";E$
790 IF E$="NO" THEN 930
800 IF E$="YES" THEN 830
810 GOSUB 1960
812 PLOT3:PLOT5:PLOT28
814 PRINT"        ";
820 GOTO 770
830 REM GET AMOUNT
835 E$="EVEN"
840 PLOT3:PLOT10:PLOT28
850 INPUT"$";H
860 IF H<=10000 THEN 890
870 GOSUB 1980
872 PLOT3:PLOT10:PLOT28
874 PRINT"        ";
880 GOTO 840
890 IF H<0 THEN 910
900 IF H=INT(H) THEN 950
910 GOSUB 2010
920 GOTO 872
930 REM NO ODD/EVEN BET
940 H=0
950 REM RED?
960 PLOT3:PLOT4:PLOT32
970 INPUT"-";C$
980 IF C$="NO" THEN 1120
990 IF C$="YES" THEN 1020
1000 GOSUB 1960
1002 PLOT3:PLOT4:PLOT32
1004 PRINT"        ";
1010 GOTO 960
1020 REM GET AMOUNT
1025 D$="RED"
1030 PLOT3:PLOT9:PLOT32
1040 INPUT"$";I
1050 IF I<=10000 THEN 1080
1060 GOSUB 1980
1062 PLOT3:PLOT9:PLOT32
1064 PRINT"        ";
1070 GOTO 1030
1080 IF I<0 THEN 1100
1090 IF I=INT(I) THEN 1240
1100 GOSUB 2010
1110 GOTO 1062
1120 REM BLACK?
1122 PLOT3:PLOT6:PLOT36
1124 INPUT"-";C$
1126 IF C$="NO" THEN 1220
1128 IF C$="YES" THEN 1136
1130 GOSUB 1960
1131 PLOT3:PLOT6:PLOT36
1132 PRINT"        ";
1134 GOTO 1122
1136 REM GET AMOUNT
1138 D$="BLACK"
1140 PLOT3:PLOT11:PLOT36
1142 INPUT"$";I
1150 IF I<=10000 THEN 1180
1160 GOSUB 1980
1162 PLOT3:PLOT11:PLOT36
1164 PRINT"        "
1170 GOTO 1140
1180 IF I<0 THEN 1200
1190 IF I=INT(I) THEN 1240
1200 GOSUB 2010
1210 GOTO 1162
1220 REM NO RED/BLACK BET
1230 I=0
1240 REM COLUMN?
1250 PLOT3:PLOT7:PLOT40
1260 INPUT"-";B1$
1270 IF B1$="NO" THEN 1490
1280 IF B1$="YES" THEN 1310
1290 GOSUB 1960
1292 PLOT3:PLOT7:PLOT40
1294 PRINT"        ";
1300 GOTO 1250
1310 PLOT3:PLOT7:PLOT40
1320 INPUT"1-2 OR 3?";B2
1330 IF B2>0 THEN IF B2<4 THEN 1400
1340 PLOT3:PLOT7:PLOT40
1345 PLOT6:PLOT79
1350 PRINT"1-2 OR 3?"
1360 FOR J1=1 TO 500:NEXT J1
1370 PLOT3:PLOT7:PLOT40
1380 PLOT6:PLOT1
1390 PRINT"          "
1395 GOTO 1310
1400 REM GET AMOUNT
1410 PLOT3:PLOT17:PLOT40
1420 INPUT" $";B8
1430 IF B8<=10000 THEN 1460
1440 GOSUB 1980
1442 PLOT3:PLOT19:PLOT40
1444 PRINT"        ";
1450 GOTO 1410
1460 IF B8<0 THEN 1480
1470 IF B8=INT(B8) THEN 1510
1480 GOSUB 2010
1482 GOTO 1442
1490 REM NO COLUMN BET
1500 B8=0
1510 REM NUMBER BET?
1520 PLOT3:PLOT7:PLOT44
1530 INPUT"-";E$
1540 IF E$="NO" THEN 1790
1550 IF E$="YES" THEN 1580
1560 GOSUB 1960
1562 PLOT3:PLOT7:PLOT44
1564 PRINT"        ";
1570 GOTO 1520
1580 PLOT3:PLOT12:PLOT44
1590 INPUT"0-36?";F
1600 IF F<0 THEN 1630
1610 IF F>36 THEN 1630
1620 IF F=INT(F) THEN 1670
1630 PLOT3:PLOT12:PLOT44
1635 PLOT6:PLOT79
1640 PRINT"0-36?"
1650 FOR J1=1 TO 500:NEXT J1
1652 PLOT3:PLOT12:PLOT44
1654 PLOT6:PLOT2
1656 PRINT"          "
1660 GOTO 1580
1670 REM GET AMOUNT
1700 PLOT3:PLOT20:PLOT44
1710 INPUT"$";G
1720 IF G<=10000 THEN 1750
1730 GOSUB 1980
1732 PLOT3:PLOT20:PLOT44
1734 PRINT"        ";
1740 GOTO 1700
1750 IF G<0 THEN 1770
1760 IF G=INT(G) THEN 2040
1770 GOSUB 2010
1780 GOTO 1732
1790 REM NO NUMBER BET
1800 G=0
1810 GOTO 2050
1960 PLOT3:PLOT0:PLOT4
1962 PLOT6:PLOT79
1964 PRINT"PLEASE!! YES OR NO";
1968 FOR J1=1 TO 500:NEXT J1
```

EVERYONE LOOKS UP TO A LEADER

Don't let competition get their leg up on your business.

Where the nation looks up!

It's important that in addition to your advertising media choices throughout the next twelve months that you also plan strategic, high impact, attention getting visuals at **Computer Shows, Trade Fairs, and Grand Openings**. Your company can have its own flying information center anywhere in the country for suprisingly little. Cal Air Advertising makes it possible for you to have your messages flown day or night over major events or population centers through its nationwide network of airborne advertising operations to give you that added visual punch. Remember when one person looks up--everyone else will too. Let them be looking up to you.

For more information contact:

Cal Air Advertising
P.O. Box 11082, Santa Ana, CA 92711 (714) 558-0644

Aircraft--Fixed wing, Helicopter, Balloon
Media--Banners & Traveling Light Displays

```
1970 PLOT3:PLOT0:PLOT4
1972 PLOT6:PLOT1
1974 PRINT"
1976 RETURN
1980 PLOT3:PLOT0:PLOT4
1982 PLOT6:PLOT79
1984 PRINT"HOUSE LIMIT IS $10,000!!";
1985 FOR J1=1 TO 500:NEXT J1
1988 PLOT3:PLOT0:PLOT4
1990 PLOT6:PLOT1
1992 PRINT"
2000 RETURN
2010 PLOT3:PLOT0:PLOT4
2012 PLOT6:PLOT79
2014 PRINT"FULL DOLLAR BETS ONLY PLEASE"
2016 FOR J1=1 TO 500:NEXT J1
2017 PLOT3:PLOT0:PLOT4
2018 PLOT6:PLOT1
2020 PRINT"
2030 RETURN
2040 REM
2050 REM
2070 T=INT(37*RND(F+H+I+B8+B2+G))
2080 REM THE NUMBER IS
2090 T1=INT(T/10)+1
2091 I1=I
2092 GOSUB 9000
2093 I=I1
2094 PLOT3:PLOT0:PLOT10:PLOT6:PLOT15
2100 ON T1 GOTO 2110,2120,2130,2140
2110 ON T+1 GOTO 2390,2210,2290,2250,2270,2230,2310,2210,2290,2250
2120 ON T-9 GOTO 2270,2350,2190,2330,2170,2370,2150,2350,2190,2210
2130 ON T-19 GOTO 2290,2250,2270,2230,2310,2210,2170,2370,2270,2350STOP
2140 ON T-29 GOTO 2190,2230,2170,2370,2150,2350,2190
2150 PRINTT;" RED,EVEN,COLUMN 1"
2160 GOTO 2400
2170 PRINTT;" RED,EVEN,COLUMN 2"
2180 GOTO 2400
2190 PRINTT;" RED,EVEN,COLUMN 3"
2200 GOTO 2400
2210 PRINTT;" RED,ODD,COLUMN 3"
2220 GOTO 2400
2230 PRINTT;" RED,ODD,COLUMN 2"
2240 GOTO 2400
2250 PRINTT;" RED,ODD,COLUMN 3"
2260 GOTO 2400
2270 PRINTT;" BLACK,EVEN,COLUMN 1"
2280 GOTO 2400
2290 PRINTT;" BLACK,EVEN,COLUMN 2"
2300 GOTO 2400
2310 PRINTT;" BLACK,EVEN,COLUMN 3"
2320 GOTO 2400
2330 PRINTT;" BLACK,ODD,COLUMN 1"
2340 GOTO 2400
2350 PRINTT;" BLACK,ODD,COLUMN 2"
2360 GOTO 2400
2370 PRINTT;" BLACK,ODD,COLUMN 3"
2380 GOTO 2400
2390 PRINT"THE NUMBER IS  0"
2400 PLOT3:PLOT60:PLOT20
2401 PLOT6:PLOT79
2402 PRINT"BETTING RESULTS"
2403 FOR G9=1 TO 500:NEXT G9
2404 PLOT6:PLOT2:PLOT3:PLOT60:PLOT20
2405 PRINT"BETTING RESULTS"
2409 IF G=0 THEN 2470
2410 IF T=F THEN 2450
2430 G=-G
2440 GOTO 2470
2450 REM
2460 G=35*G
2470 IF H=0 THEN 2660
2480 IF T=0 THEN 2570
2490 IF B$="EVEN" THEN 2540
2500 FOR X=1 TO 35 STEP 2
2510 IF T=X THEN 2610
2520 NEXT X
2530 GOTO 2570
2540 FOR X1=2 TO 36 STEP 2
2550 IF T=X1 THEN 2610
2560 NEXT X1
2570 REM
2590 H=-H
2610 REM
2620 IF B$="EVEN" THEN 2640
2630 PLOT3:PLOT63:PLOT24
2635 GOTO 2642
2640 PLOT3:PLOT64:PLOT28
2642 IF H<0 THEN 2646
2644 PLOT6:PLOT2
2645 GOTO 2650
2646 PLOT6:PLOT1
2650 PRINT" $";H
2660 IF I=0 THEN 2940
2670 IF T=0 THEN 2840
2680 FOR A1=1 TO 9 STEP 2
2690 IF T=A1 THEN 2830
2700 NEXT A1
2710 FOR A2=12 TO 18 STEP 2
2720 IF T=A2 THEN 2830
2730 NEXT A2
2740 FOR A3=19 TO 25 STEP 2
2750 IF T=A3 THEN 2830
2760 NEXT A3
2770 FOR A4=30 TO 36 STEP 2
2780 IF T=A4 THEN 2830
2790 NEXT A4
2800 IF T=26 THEN 2830
2810 IF D$="BLACK" THEN 2860
2820 GOTO 2840
2830 IF D$="RED" THEN 2880
2840 REM
2860 I=-I
2880 REM
2890 IF D$="BLACK" THEN 2920
2900 PLOT3:PLOT63:PLOT32
2910 GOTO 2930
2920 PLOT3:PLOT65:PLOT36
2930 IF I<0 THEN 2936
2932 PLOT6:PLOT2
2934 GOTO 2938
2936 PLOT6:PLOT1
2938 PRINT" $";I
2940 IF B8=0 THEN 3212
2950 IF T=0 THEN 3160
2960 FOR B3=1 TO 34 STEP 3
2970 IF T=B3 THEN 3050
2980 NEXT B3
2990 FOR B4=2 TO 35 STEP 3
3000 IF T=B4 THEN 3070
3010 NEXT B4
3020 FOR B5=3 TO 36 STEP 3
3030 IF T=B5 THEN 3090
3040 NEXT B5
3050 IF B2=1 THEN 3110
3060 IF B2<>1 THEN 3160
3070 IF B2=2 THEN 3110
3080 IF B2<>2 THEN 3160
3090 IF B2=3 THEN 3110
3100 IF B2<>3 THEN 3160
3110 REM
3130 REM
3140 B8=2*B8
3150 GOTO 3200
3160 REM
3190 B8=-B8
3200 IF B8<0 THEN 3206
3202 PLOT6:PLOT2
3204 GOTO 3208
3206 PLOT6:PLOT1
3208 PLOT3:PLOT66:PLOT40
3210 PRINT" $";B8
3212 IF G=0 THEN 3220
3213 IF G<0 THEN 3216
3214 PLOT6:PLOT2
3215 GOTO 3218
3216 PLOT6:PLOT1
3218 PLOT3:PLOT66:PLOT44
3219 PRINT" $";G
3220 PLOT3:PLOT68:PLOT10
3230 PRINT"
3235 K9=K9+G+H+I+B8
3240 IF K9<0 THEN 3270
3250 PLOT6:PLOT2
3260 GOTO 3280
3270 PLOT6:PLOT1
3280 PLOT3:PLOT68:PLOT10
3290 PRINT" $";K9
4990 FOR I1=1 TO 2000:NEXT I1
4995 GOTO 500
4999 END
5000 REM DRAW THE BOARD
5010 PLOT6:PLOT7:PLOT15:PLOT12
5040 PLOT6:PLOT56
5050 PLOT3:PLOT35:PLOT47
5060 FOR I=1 TO 11:PLOT32:NEXT I
5080 FOR I=44 TO 20 STEP -2
5090 PLOT3:PLOT35:PLOT I
5100 FOR J=1 TO 11:PLOT32:NEXT J
5120 NEXT I
5140 PLOT3:PLOT35:PLOT17
5160 FOR I=1 TO 11:PLOT32:NEXT I
5180 FOR I=20 TO 44 STEP 12
5200 PLOT3:PLOT29:PLOT I
5220 FOR J=1 TO 23:PLOT32:NEXT J
5230 NEXT I
5235 FOR I=28 TO 52 STEP 24 :REM START VERTICAL
5240 FOR J=44 TO 20 STEP-1
5260 PLOT3:PLOT I:PLOT J
5280 PLOT32:NEXT J
5300 NEXT I
5320 FOR I=34 TO 46 STEP 12
5340 FOR J=17 TO 47
5360 PLOT3:PLOT I:PLOT J
5380 PLOT32:NEXT J:NEXT I
5400 FOR I=30 TO 42 STEP 4
5420 FOR J=20 TO 47
5440 PLOT3:PLOT I:PLOT J
5460 PLOT32:NEXT J:NEXT I
5500 REM LABEL THE BOARD
5520 PLOT6:PLOT20
5540 FOR I=29 TO 47 STEP 18
5560 FOR J=21 TO 31
5580 PLOT3:PLOTI:PLOTJ
5590 FOR K=1 TO 5:PLOT32:NEXT K
5600 NEXT J:NEXT I
5610 FOR I=18 TO 19
5615 PLOT3:PLOT35:PLOTI
5620 FOR J=1 TO 11
5640 PLOT32:NEXT J:NEXT I
5650 FOR I=45 TO 46
5660 FOR J=35 TO 43 STEP 4
5670 PLOT3:PLOTJ:PLOTI
5680 FOR K=1 TO 3:PLOT32:NEXT K
5690 NEXT J:NEXT I
5700 PLOT3:PLOT40:PLOT19:PRINT"0"
5710 PLOT3:PLOT30:PLOT26:PRINT"EVEN"
5720 PLOT3:PLOT48:PLOT26:PRINT"ODD"
5730 FOR I=35 TO 43 STEP 4
5740 PLOT3:PLOTI:PLOT45
5750 PRINT"COL";NEXT I
5760 PLOT3:PLOT36:PLOT46
5765 K=0
5770 FOR I=35 TO 43 STEP 4
5780 K=K+1
5790 PLOT3:PLOTI:PLOT46
5800 PRINTK
5810 NEXT I
5820 PLOT6:PLOT7
5830 PLOT3:PLOT29:PLOT36
5840 PRINT"BLACK"
5850 PLOT6:PLOT15
5860 FOR I=33 TO 42
5870 PLOT3:PLOT47:PLOTI
5880 FOR J=1 TO 5:PLOT32:NEXTJ
5890 NEXT I
5900 PLOT3:PLOT40:PLOT38:PRINT"RED"
5910 PLOT3:PLOT35:PLOT21:PRINT" 1 "
5912 PLOT3:PLOT43:PLOT21:PRINT" 3 "
5914 PLOT3:PLOT39:PLOT23:PRINT" 5 "
5916 PLOT3:PLOT35:PLOT25:PRINT" 7 "
5918 PLOT3:PLOT43:PLOT25:PRINT" 9 "
5920 PLOT3:PLOT43:PLOT27:PRINT"12 "
5922 PLOT3:PLOT39:PLOT29:PRINT"14 "
5924 PLOT3:PLOT35:PLOT31:PRINT"16 "
5926 PLOT3:PLOT43:PLOT31:PRINT"18 "
5928 PLOT3:PLOT35:PLOT33:PRINT"19 "
5930 PLOT3:PLOT43:PLOT33:PRINT"21 "
5932 PLOT3:PLOT39:PLOT35:PRINT"23 "
5934 PLOT3:PLOT35:PLOT37:PRINT"25 "
5936 PLOT3:PLOT39:PLOT37:PRINT"27 "
5938 PLOT3:PLOT43:PLOT39:PRINT"30 "
5940 PLOT3:PLOT39:PLOT41:PRINT"32 "
5942 PLOT3:PLOT35:PLOT43:PRINT"34 "
5944 PLOT3:PLOT43:PLOT43:PRINT"36 "
5946 PLOT6:PLOT7
5950 PLOT3:PLOT39:PLOT21:PRINT" 2"
5952 PLOT3:PLOT35:PLOT23:PRINT" 4"
5954 PLOT3:PLOT43:PLOT23:PRINT" 6"
5956 PLOT3:PLOT39:PLOT25:PRINT" 8"
```

```
5958 PLOT3:PLOT35:PLOT27:PRINT"10"
5960 PLOT3:PLOT39:PLOT27:PRINT"11"
5898 PLOT3:PLOT35:PLOT29:PRINT"13"
5964 PLOT3:PLOT43:PLOT29:PRINT"15"
5966 PLOT3:PLOT39:PLOT31:PRINT"17"
5968 PLOT3:PLOT39:PLOT33:PRINT"20"
5970 PLOT3:PLOT35:PLOT35:PRINT"22"
5972 PLOT2:PLOT43:PLOT35:PRINT"24"
5974 PLOT3:PLOT43:PLOT37:PRINT"27"
5975 PLOT3:PLOT35:PLOT39:PRINT"28"

5976 PLOT3:PLOT39:PLOT39:PRINT"29"
5978 PLOT3:PLOT35:PLOT41:PRINT"31"
5980 PLOT3:PLOT43:PLOT41:PRINT"33"
5982 PLOT3:PLOT39:PLOT43:PRINT"35"
5990 REM END OF BOARD
6000 REM PLACE TEXT
6010 PLOT6:PLOT2
6015 PLOT146
37399 3:PLOT0:PLOT20
6040 PRINT"PLACE YOUR BETS":PRINT

6050 PRINT"ODD":PRINT
6060 PRINT"EVEN":PRINT
6070 PRINT"RED":PRINT
6080 PRINT"BLACK":PRINT
6090 PRINT"COLUMN":PRINT
6100 PRINT"NUMBER"
6110 PLOT3:PLOT60:PLOT20
6120 PRINT"BETTING RESULTS"
6130 PLOT3:PLOT60:PLOT24
6140 PRINT"ODD"
6150 PLOT3:PLOT60:PLOT28
6160 PRINT"EVEN"
6170 PLOT3:PLOT60:PLOT32
6180 PRINT"RED"
6190 PLOT3:PLOT60:PLOT36
6200 PRINT"BLACK"
6210 PLOT3:PLOT60:PLOT40
6220 PRINT"COLUMN"
6222 PLOT3:PLOT60:PLOT44
6224 PRINT"NUMBER"
6230 PLOT3:PLOT0:PLOT10
6250 PLOT3:PLOT60:PLOT10
6260 PRINT"BALANCE $0"
6270 PLOT15:PLOT6:PLOT7
6280 RETURN
6300 REM CLEAR THE BET AREA
6310 PLOT6:PLOT7
6320 PLOT3:PLOT3:PLOT24
6330 PRINT"                    "
6340 PLOT3:PLOT4:PLOT28
6350 PRINT"                    "
6360 PLOT3:PLOT3:PLOT32
6370 PRINT"                    "
6380 PLOT3:PLOT5:PLOT36
6390 PRINT"                    "
6400 PLOT3:PLOT6:PLOT40
6410 PRINT"                    "
6420 PLOT3:PLOT5:PLOT44
6430 PRINT"                    "
6440 RETURN
6500 REM CLEAR THE RESULTS AREA
6510 PLOT6:PLOT7
6520 PLOT3:PLOT63:PLOT24
6530 PRINT"                    "
6540 PLOT3:PLOT64:PLOT28
6550 PRINT"                    "
6560 PLOT3:PLOT63:PLOT32
6570 PRINT"                    "
6580 PLOT3:PLOT65:PLOT36
6590 PRINT"                    "
6600 PLOT3:PLOT66:PLOT40
6610 PRINT"                    "
6620 PLOT3:PLOT66:PLOT44
6630 PRINT"                    "
6640 RETURN
9000 REM SPIN THE BALL COUNTER CLOCKWISE
9005 PLOT3:PLOT80:PLOT0
9010 PLOT2:PLOT253:PLOTX:PLOTY
9020 FOR I=1 TO 2
9030 FOR K=1 TO K4
9035 REM PLOT THE WHITE BALL
9040 PLOT255:PLOT6:PLOT7:PLOT2:PLOT253
9050 PLOT X1(K):PLOT Y1(K)
9060 REM PLOT THE BLACK BALL
9070 PLOT255:PLOT6:PLOT0:PLOT2:PLOT253
9080 PLOT X1(K):PLOT Y1(K)
9090 NEXT K:NEXT I
9095 REM END OF CCW SPIN
9100 REM SPIN ONCE CW
9110 PLOT255:PLOT3:PLOT80:PLOT0
9112 PLOT6:PLOT7
9115 PLOT2:PLOT253:PLOTX:PLOTY
9120 FOR I=K4 TO 1 STEP-1
9140 PLOT255:PLOT6:PLOT7:PLOT2:PLOT253
9160 PLOTX2(I)
9170 PLOTY2(I)
9180 PLOT255:PLOT6:PLOT0:PLOT2:PLOT253
9190 PLOTX2(I)
9200 PLOTY2(I)
9210 NEXT I
9220 PLOT255:PLOT6:PLOT7
9230 REM END OF SPIN
9240 RETURN
10000 REM CALCULATE THE PATH OF THE BALL
10010 DIM X1(64),Y1(64)
10015 DIM X2(64),Y2(64)
10020 S1=10:K1=0:K2=158:K3=S1:K4=0
10030 XX=20:YY=30
10035 X3=17:Y3=27
10040 REM CENTER OF WHEEL
10050 X=80:Y=160
10060 FOR KK=1 TO 4
10070 IF KK<>2 THEN 10090
10080 XX=-XX:K1=158:K2=0:K3=-S1
10085 X3=-X3
10090 IF KK<>3 THEN 10110
10100 YY=-YY:K1=0:K2=158:K3=S1
10105 Y3=-Y3
10110 IF KK<>4 THEN 10130
10120 XX=-XX:K1=158:K2=0:K3=-S1
10125 X3=-X3
10130 FOR I=K1 TO K2 STEP K3
10140 A=I*.01
10150 K4=K4+1
10160 X1(K4)=X+XX*COS(A)
10170 Y1(K4)=Y+YY*SIN(A)
10172 X2(K4)=X+X3*COS(A)
10174 Y2(K4)=Y+Y3*SIN(A)
10180 NEXT I
10190 NEXT KK
10192 RETURN
10195 REM END OF CALCULATION
```

MISFIT

By Bruce A. Scott

Misfit is a new computer game which can involve from one to five players. To score, a player must recognize a relationship between four items which the machine presents.

Speed is just as important as accuracy in this game since recognizing the misfits and the similarities between the items is important only if you get your answer in first.

Each time the computer reveals a new term, the players have the option of striking their personal marker key or passing. If all of them pass, one of them hits the return key and the computer reveals the next word in the set. The first player to spot the misfit strikes his key and tells the computer which word does not fit with the others.

If he is right, the computer awards him a letter and asks him to NAB or STAB. He stabs by typing something that he thinks the remaining three terms have in common. He earns one letter if he spots the significant similarity. The computer reveals the next word in the set if he is wrong.

The player dares another player to state the similarity by typing NAB and the player's name or chosen key. If the nabbed player doesn't know the similarity, the challenger wins two letters. The challenged player wins two letters if he does recognize what the three terms have in common. The first player to earn all of the letters in *MISFIT* wins the game.

The program has six sets of terms as it is printed here. You can replace any of them or add more sets easily. Lines 160 throuh 210 are used for this purpose. The format, TERM1, TERM2, TERM3, TERM4, MISFIT, and SIMILARITY is used throughout.

Let me offer you one more set and tell you how to modify the program if you want to use it. It is not my purpose to help you increase your set count from six to seven. I am trying to show you how to modify the program to include the sets of terms that you want to use.

The three numbers 1, 2, and 4 are all base 2 numbers. 3 is not a base 2 number, but the numbers 1, 2, 3, and 4 look like they belong together. It takes a little imagination to spot the misfit 3 and recognize what the other three have in common.

If you want to use this set to replace the set on line 200, type 200 DATA 1,2,3,4,3,BASE 2. First type the four terms, then the misfit term, finally the significant similarity.

If you want to add this new set to the data base without losing one of the other sets, pick some line number between 160 and 220 that isn't in use and type 201 DATA 1,2,3,4,3,BASE 2. You also have to make the following changes to tell the computer that you have changed the quantity of available sets.

LINE TERM	CHANGED TO
120 M$(6,6)	M$(#,6)
150 FORV = 1T06	FORV = 1T0#
280 V2 = INT(RND(0)*6 + 1)	V2 = INT(RND(0)*# + 1)

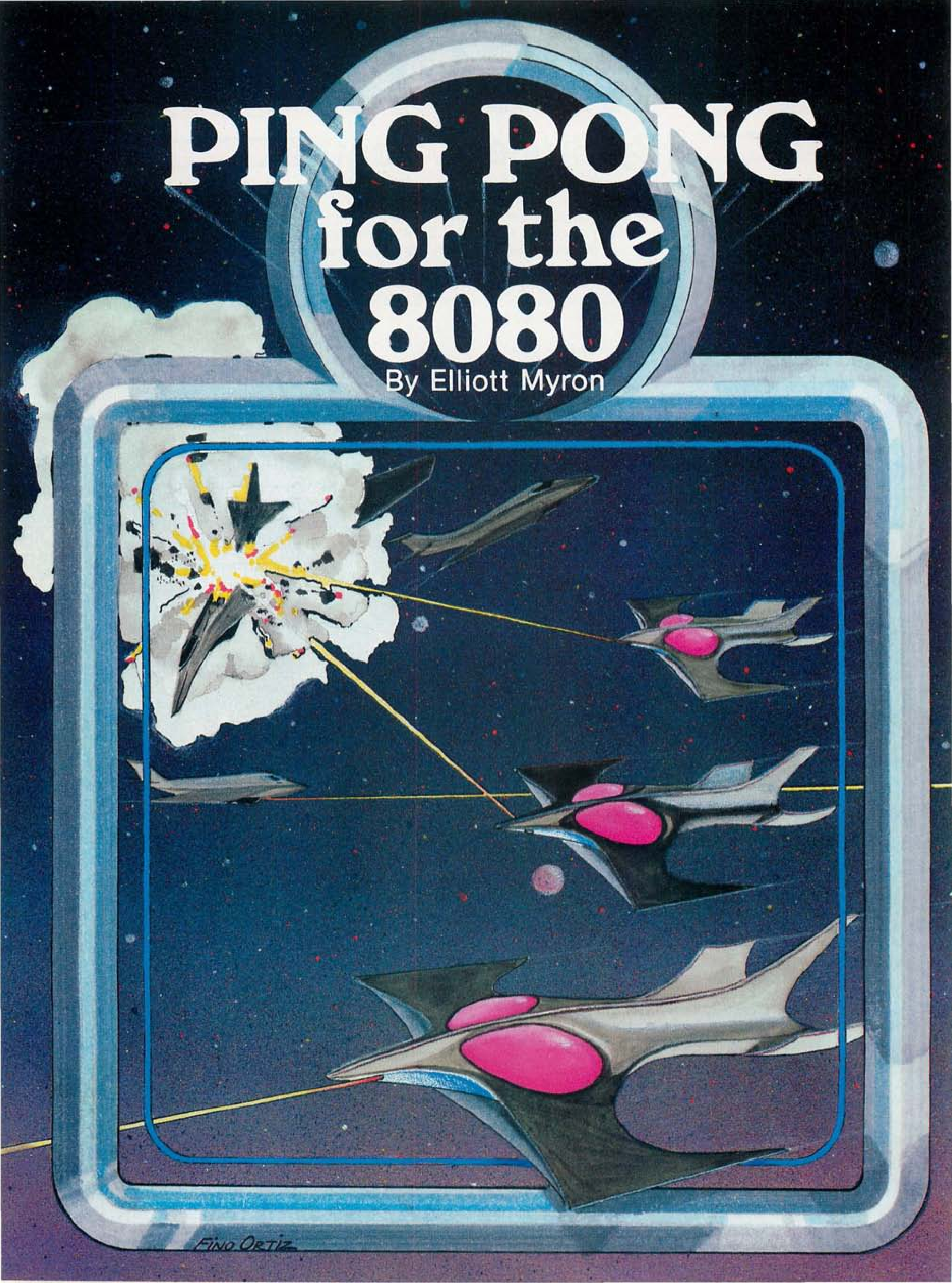
Use the quantity of available sets in place of the # symbol when you make the changes shown above.

Your computer should be able to run this program if it speaks BASIC and handles string literals. It was written on a Digital Equipment Corporation PDP 11-45/E that had a BASIC-PLUS compiler. I have tried to avoid program statements that are unique to BASIC-PLUS. I hope that I have been successful and that you will be able to make the changes that invariably exist between different BASIC compilers.□

```
110 !CØMPUTER GAME FØR 1 TØ 5 PLAYERS
120 DIMP$(5,2),M$(6,6),T$(5),S$(5,8)
130 FØRH=1TØ8:READS$(0,H):NEXTH
140 DATA'* ','M ','I ','S ','F ','I ','T ','* '
150 FØRV=1TØ6:FØRH=1TØ6:READM$(V,H):NEXTH:NEXTV
160 DATA FRENCH,ITALIAN,SPANISH,RUSSIAN,SPANISH,DRESSING
170 DATA'RED','WHITE','BLUE','GREEN','BLUE','MEXICAN FLAG'
180 DATA'HURØN','ØNTARIØ','MICHIGAN','ERIE','MICHIGAN','CANADA'
190 DATA'RED','WHITE','BLUE','YELLØW','WHITE','PRIMARY CØLØRS'
200 DATA'FRENCH','ITALIAN','SPANISH','RUSSIAN',RUSSIAN,LATIN
210 DATA RED, WHITE,BLUE,GREEN,WHITE,TELEVISIØN TUBE
220 INPUT'HØW MANY PLAYERS';T1:FØRV1=1TØT1:PRINT:PRINT'PLAYER';V1;
230 INPUT'WHAT IS YØUR NAME AND WHICH KEY IS YØURS';P$(V1,1),P$(V1,2)
240 PRINT'THANK YØU ';P$(V1,1):NEXTV1
250 PRINT'I WILL SHØW YØU FØUR TERMS ØNE AT A TIME.  ØNE ØF THE'
260 PRINT'FØUR TERMS IS A MISFIT.  HIT YØUR KEY WHEN YØU SPØT IT.
270 PRINT
280 H1=1:V2=INT(RND(0)*6+1):FØRI=1TØ4:T$(I)='********':NEXTI
290 IFR=8THEN560
300 T$(H1)=M$(V2,H1):FØRI=1TØ4:PRINTT$(I):NEXTI:INPUTR$
310 IFR$=' 'THEN520
320 FØRI1=1TØT1:IFP$(I1,2)=R$THEN340
330 NEXTI1:INPUT'KEY ERRØR TRY AGAIN';R$:GØTØ310
340 PRINTP$(I1,1);:INPUTR$:IFR$=M$(V2,5)THEN360
350 PRINT'WRØNG! ';R$;' IS NØT THE MISFIT.':GØTØ520
360 FØRV=1TØ4:IFM$(V2,V)=R$THENT$(V)=' '
370 NEXTV:PRINT'RIGHT! ';P$(I1,1);' GETS ';:GØSUB530
380 INPUT'DØ YØU NAB ØR STAB ';R$:IFLEFT(R$,3)='NAB'THEN410
390 IFR$<>M$(V2,6)THENPRINT'WRØNG!!':GØTØ520
400 PRINT'THATS THE WAY. ';P$(I1,1);' GETS ';:GØSUB530:GØTØ500
410 R$=RIGHT(R$,4):IFR$=' 'THEN450
420 IFLEFT(R$,1)=' 'THENR$=RIGHT(R$,2):GØTØ420
430 FØRI3=1TØT1:FØRI2=1TØ2:IFR$=P$(I3,I2)THEN460
440 NEXTI2:NEXTI3
450 INPUT'WHØ';R$:GØTØ420
460 PRINTP$(I3,1);' WHAT DØ THE TERMS':FØRI4=1TØ4:PRINTT$(I4):NEXTI4
470 INPUT'HAVE IN CØMMØN';R$:IFR$=M$(V2,6)THENI1=I3:GØTØ490
480 PRINT'SØRRY ';P$(I3,1);' ';
490 PRINTP$(I1,1);' GETS ';:GØSUB530:GØSUB530
500 PRINT'THE SCØRE IS NØW':FØRI5=1TØT1:PRINTP$(I5,1);':';,
510 FØRI6=1TØ8:PRINTS$(I5,I6);:NEXTI6:PRINT:NEXTI5:GØTØ280
520 H1=H1+1:GØTØ300
530 R=VAL(S$(I1,0))+1:PRINTS$(0,R);:S$(I1,R)=S$(0,R)
540 IFR=8THENPRINTP$(I1,1);' WINS.';:GØTØ500
550 S$(I1,0)=NUM$(R):RETURN
560 END
```

PING PONG for the 8080

By Elliott Myron



In a previous article entitled "Crazy Ball" by this author for INTERFACE AGE (August 1977), the idea of using the computer with CRT display for playing video games was discussed. This article will examine the possibility of producing a very inexpensive pair of control boxes for two players to enable the playing of video games such as 'Ping Pong'. Later in this article you will find a listing of the game of 'Ping Pong' which the author uses on his own system. The program is written in BASIC (MITS) and requires the use of a CRT interface board such as the PolyMorphic VTI which has graphic characters.

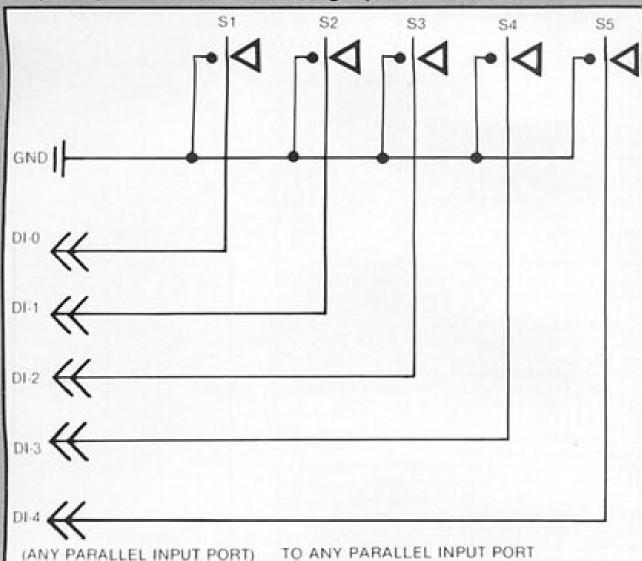


Figure 1. Schematic for Control Box

In most video games it is only necessary to be able to execute the following commands: up, down, left, and right. In some instances a fifth command may be issued such as 'fire a missile' or 'serve the ball'. We will construct a small control box with similar commands. It will be outfitted with five momentary switches as is shown in the switch layout in Figure 2. The action which is initiated by each switch may be decided upon at the time of programming, but for the game of Pong, only three of these switches will be used to obtain the following movements: up, down, and serve. This will be discussed in more detail later.

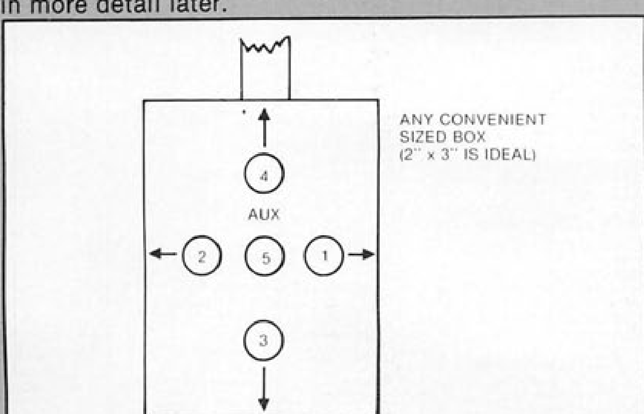


Figure 2. Switch Layout

To construct the control boxes you must have the following items:

1) Five momentary switches (normally open) for each control box.
2) A suitable enclosure (2" x 3" x 1" deep is ideal) for each control you wish to manufacture.
3) Six conductor cables (approximately 5 or 6 feet per control box).
4) A connector plug and jack with multiple pins (1 25-pin connector would be more than sufficient for as many as four control boxes).
5) A parallel input port for each control box.

Using the configuration shown in Figure 2, install the switches in the box; then connect them as shown in Figure 1.

S1 connects to data bit 0 (DI-0), S2 connects to data bit 1 (DI-1) and so on. The sixth wire connects to ground of your computer and is bussed to one contact of each of the five switches.

The method of operation is as follows. Let us say that you have connected your control box to input port #1. If no buttons are pushed, an input from port #1 would result in all eight bits being in a high state. This is the number 255 (decimal) or FF (HEX). Pushing any one button causes the associated bit to drop down to a logic zero. For example, suppose that switch S1 was pushed. This action would result in the following binary pattern: 11111110 (S1 is connected to DI-0 which is the least significant bit). The number which is received by the computer in port #1 is 254 (decimal) or FE (HEX). The following table may be of use:

SWITCH	BINARY PATTERN	DECIMAL EQUIVALENT	HEX EQUIVALENT
S1	11111110	254	FE
S2	11111101	253	FD
S3	11111011	251	FB
S4	11110111	247	F7
S5	11101111	239	EF
NONE	11111111	255	FF

For the purpose of programming, the programmer may wish to use the compliment of the input.

In BASIC, a program which tests the input of a control box may look something like the following:

```
100 J = INP(1):REM   INPUT FROM THE CONTROL BOX ON PORT #1
105 IF J = 254 THEN  200:REM PROCESS A MOVE RIGHT COMMAND
110 IF J = 253 THEN  300:REM PROCESS A MOVE LEFT COMMAND
115 IF J = 251 THEN  400:REM PROCESS A MOVE DOWN COMMAND
120 IF J = 247 THEN  500:REM PROCESS A MOVE UP COMMAND
125 IF J = 239 THEN  600:REM PROCESS AN AUXILIARY COMMAND
                     SUCH AS SERVE
130 GO TO 100: REM LOOP FOR A COMMAND
```

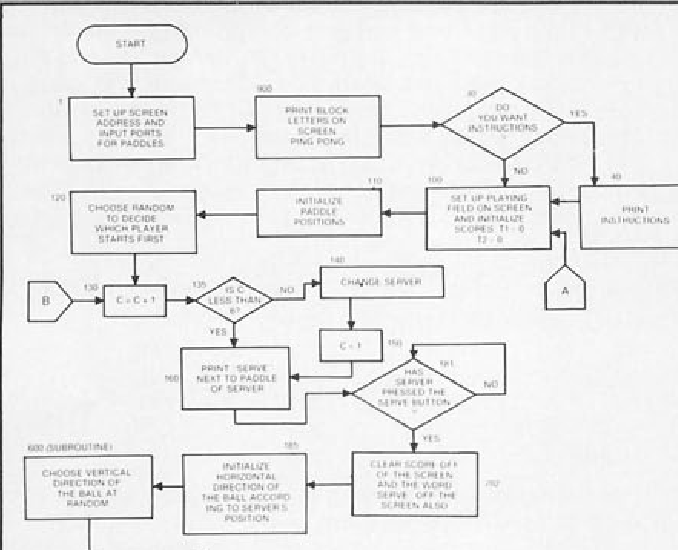
The routines which would be found at 200, 300, etc. are routines that would move a cursor (or other similar character) around the screen. The routines can be as involved as the programmer desires; however, care should be taken to make the routines as efficient as possible with respect to execution time. When the game of Pong is played, the ball moves at a relatively slow speed because we are using BASIC which is considerably slower than a machine language program would be.

To use the BASIC program to play Pong, you must have a CRT interface which allows poking characters (graphic) directly into memory. An example would be the PolyMorphic VTI board with 16 lines by 64 characters per line. The program is set for the PolyMorphic board being addressed at 7C00H (31744 decimal). You may modify the program to suit your present display address by changing the first line so that P = your CRT address (in decimal). The program is also set to use two control boxes — one on port #1 and the other on port #2. You may also change this set-up by changing line 1 so that I1 and I2 are equal to the port numbers you wish to use. In addition, the BASIC which I use allows a clear screen command by printing CHR$(5). If your BASIC uses a different number, substitute that number for the 5 in CS = 5 found in line 1. You may also substitute the following for a clear screen in lines 40, 101, 506, 900, and 910:

```
(line #)  GO SUB 1000
   1000   FOR CS = 1 TO 16:PRINT:NEXT CS:RETURN
```

The above has the effect of printing 16 returns to clear the screen. If you find that the program uses too much of your memory, you may delete line 20 and lines 800 through 910. These lines are used to poke the words PING and PONG in block letters at the beginning of the program.

Several of my students have developed many fascinating programs which use the control boxes. These games include 'CHASE', 'LOGAN'S RUN', 'OBSTACLE RACE', 'ATTACK', and many more. If there is enough interest for this type of program, we will make them available for publication and distribution.☐



Flow Chart for Ping Pong

PROGRAM LISTING

```
0 REM ******************** PING PONG ************************
1 P=31744:CS=31:NS=960:I1=1:I2=2
2 DIM SR(5):FORI=1TO5:READSR(I):NEXT
3 DATA 211,147,210,214,177
10 T1=0:T2=0
20 GOTO800
30 INPUT"DO YOU WANT INSTRUCTIONS";A$
31 IF LEFT$(A$,1)="Y" THEN 40
32 GOTO100
40 PRINTCHR$(CS)
41 PRINT"PLAYER #1 USES THE CONTROL BOX ON PORT #"I1":"
42 PRINT"PLAYER #2 USES THE CONTROL BOX ON PORT #"I2":"
43 PRINT:PRINT"USE THE TOP BUTTON TO MOVE YOUR PADDLE UP, AND THE"
44 PRINT"BOTTOM BUTTON TO MOVE THE PADDLE DOWN.  TO SERVE, PUSH"
45 PRINT"THE MIDDLE BUTTON.  FIFTEEN POINTS IS A GAME."
50 PRINT:PRINT:INPUT"READY";A$
51 IF LEFT$(A$,1)<>"Y" THEN 50
100 REM   SET UP FIELD
101 PRINTCHR$(CS):POKEP+54,160
102 FORI=0TO960 STEP64
103 POKEP+I,63:POKEP+63+I,63:NEXT
104 FORI=31 TO 991STEP64
105 POKE P+I,7:NEXT
106 FORI=0TO63
107 POKEP+I,54:POKENS+P+I,27:NEXT
110 REM   SET UP PADDLES
113 P(1)=P+456:P(2)=P+569
115 POKE P(1),0:POKE P(2),0
120 R=INT(RND(1)*2)+1
130 C=C+1
135 IFC<6 THEN 160: REM      TO PRINT "SERVE"
140 IFR=1 THEN R=2:GOTO150
145 R=1
150 C=1
160 IFR=1THEN174:DB=P(1)+1:GOTO174
170 DB=P(2)-1:FORI=1TO5
172 POKE P(2)+I,SR(I):NEXTI:PT=I2:SV=239:REM PLAYER 2 SERVES
173 GOTO180:REM      TO SERVE
174 FORI=5TO1STEP-1
175 POKE P(1)-I,SR(6-I):NEXT:PT=I1:SV=239:REM PLAYER 1 SERVES
180 SYM=45:DA=PEEK(DB+F1):REM INITIALIZE BALL SYMBOL & WAIT FOR SERVE
181 IFINP(PT)=SV THEN DA=160:GOTO702
182 GOTO181
183 FORI=1TO5
184 POKEP(1)-I,160:POKEP(2)+I,160:NEXT
185 IFR=1THENF1=1
186 IFR=2 THEN F1=-1
190 REM GET F2
191 GOSUB600
200 REM    START TO MOVE BALL AND PADDLES
201 IF INP(I2)=247 THEN 204:REM PADDLE UP
202 IF INP(I2)=251 THEN 206:REM PADDLE DOWN
203 GOTO210
204 IF PEEK(P(2)-64)<>160 THEN 210
205 POKEP(2),160:P(2)=P(2)-64:GOTO210
206 IF PEEK(P(2)+64)<>160 THEN 210
207 POKEP(2),160:P(2)=P(2)+64
210 IF INP(I1)=247 THEN 213:REM PADDLE UP
211 IF INP(I1)=251 THEN 215:REM PADDLE DOWN
212 GOTO220
213 IFPEEK(P(1)-64)<>160 THEN 220
214 POKEP(1),160:P(1)=P(1)-64:GOTO220
215 IFPEEK(P(1)+64)<>160 THEN 220
216 POKEP(1),160:P(1)=P(1)+64
220 POKEP(1),0:POKEP(2),0
230 J=INT(RND(1)*7+2)
231 FORN=1TOJ:IFN=J AND F2<>0 THEN 300
235 IFPEEK(DB+F1)=160 THEN 245
240 IF PEEK(DB+F1)=7 THEN 245
241 IF PEEK(DB+F1)=63 THEN 500
242 IF PEEK(DB+F1)=0 THEN F1=F1*(-1)
245 POKEDB,DA:DA=PEEK(DB+F1):DB=DB+F1:POKE(DB),SYM
250 IFPEEK(DB+F1)=0 THEN F1=F1*(-1):GOSUB600:REM HIT PADDLE
255 NEXT
260 GOTO200
300 POKEDB,DA:IF SYM=45 AND F2=1 THEN SYM=27:GOTO330
301 IF SYM=45 AND F2=-1 THEN SYM=54:GOTO330
302 IF SYM=27 AND F2=1 THEN SYM=54:POKEDB,DA:DB=DB-64:GOTO310
303 IF SYM=54 AND F2=1 THEN SYM=45:GOTO330
304 IF SYM=54 ANDF2=-1 THEN SYM=27:POKEDB,DA:DB=DB+64:GOTO315
305 IF SYM=27 AND F2=-1 THEN SYM=45:GOTO330
310 IF DB<P+64 THEN F2=-1:DB=DB+64:GOTO330
311 GOTO330
315 IF DB>P+959 THEN F2=1:DB=DB-64:GOTO330
316 GOTO330
330 GOTO235
500 IF F1=1 THEN T1=T1+1:GOTO502
501 T2=T2+1
502 POKEP(1),160:POKEP(2),160
503 P(1)=P+454:P(2)=P+569:POKEP(1),0:POKEP(2),0
504 IF T1=15 OR T2=15 THEN 506
505 POKEDB,DA:GOTO700
506 PRINTCHR$(CS):IFT1=15 THEN PRINT"PLAYER #1 WINS THIS TIME!":GOTO510
507 PRINT"PLAYER #2 WINS THIS TIME!"
509 PRINT"THE SCORE WAS"T2"TO"T1:GOTO511
510 PRINT"THE SCORE WAS"T1"TO"T2
511 PRINT:PRINT:INPUT"ANOTHER GAME";A$
512 IF LEFT$(A$,1)="Y" THEN T1=0:T2=0:GOTO100
513 END
600 F3=F2
601 F2=INT(RND(1)+.5)-INT(RND(2)+.5)
603 IF F2=F3 THEN 601
604 RETURN
700 A$=STR$(T1):H=P+54:GOSUB710
701 A$=STR$(T2):H=P+124:GOSUB710:GOTO130
702 REM    CLEAR SCORE OFF OF SCREEN
703 POKE P+65,160:POKEP+67,160:POKEP+126,160:POKEP+127,160
704 GOTO183
710 FOR G=2 TO LEN(A$)
711 F=VAL(MID$(A$,G,1))
712 POKE H+G,F+176:NEXT
713 RETURN
800 REM  DATA FOR P
801 DATA 67,68,69,70,71,72,73,74,75,76,77,78,131,132,141,142
802 DATA 195,196,197,198,199,200,201,202,203,204,205,206,259,260
803 DATA 323,324,387,388
810 REM DATA FOR  I
811 DATA 211,212,213,214,215,216,217,218,219,220,221,222,280,281
812 DATA 344,345,408,409,472,473
813 DATA 531,532,533,534,535,536,537,538,539,540,541,542
820 REM DATA FOR  N
821 DATA 355,356,357,365,366,419,420,421,422,429,430
822 DATA 483,484,485,486,487,488,493,494,547,548
823 DATA 552,553,554,555,556,557,558,611,612,618,619,620,621,622
824 DATA 675,676,684,685,686
830 REM DATA FOR G
831 DATA499,500,501,502,503,504,505,506,507,508,509,510,563,564
832 DATA627,628,691,692,698,699,700,701,702,755,756,765,766
833 DATA819,820,821,822,823,824,825,826,827,828,829,830
840 REM DATA TO BLANK OUT STEM OF I
841 DATA 280,281,344,345,408,409,472,473
850 REM DATA TO ADD SIDES OF O.
851 DATA276,275,283,285,339,340,349,350,403,404
852 DATA 413,414,457,458,477,478
900 PRINTCHR$(CS):POKEP+64,160
901 FORI=1TO145:READA:POKEP+A,0:NEXT
902 FORI=1TO500:NEXT
903 FORI=1TO8:READA:POKEP+A,160:NEXT
904 FORI=1TO16:READA:POKEP+A,0:NEXT
905 FORI=1TO1000:NEXT
910 PRINTCHR$(CS):GOTO30
OK
```

Calculator Consideration Survey

By John D. Hirsch

In early 1976 Dr. Pierre Brind'Amour, a professor of classical studies at the University of Ottawa, was browsing through a weighty tomes of collected Latin inscriptions when he came upon a reference to an inscription from the Etruscan sarcophagus of a certain Salvius. The inscription said that Salvius had died on the fifth day before the Ides of March, on the third day of the (new) moon, during the period when Piso and M. Acilius were consuls.

Dr. Brind'Amour had just spent three years writing a book on the pre-Julian calendar (the Roman calendar prior to the adoption of the Julian calendar by Caesar in 45 B.C.). The date of Salvius' death could be established as October 11, 67 B.C. According to Dr. Brind'Amour's manuscript, this date was equivalent to September 10th by the Julian calendar, had it existed in 67 B.C. This was the first chance to check the validity of his theory. Dr. Brind'Amour rushed to Goldstine's tables of New and Full-Moons and found that September 8th Julian was a new moon date, so September 10th was the third day after the new moon and his theory was initially confirmed. In his own words, "Archimedes out of his bath had not felt any lighter."

But a few moments later the Canadian scholar realized that he had been looking at the year -67 in Goldstine's tables, while 67 B.C. was really astronomically equivalent to -66. He turned to the year -66 in the tables and found that September 8th was *not* a new moon date. His theory, developed after three years of hard work, had failed in its first test, and the printing of his book had to be halted.

Dr. Brind'Amour decided to buy a programmable calculator as an aid in reworking his theories.

His revised theories resulted in a new manuscript of over 1,000 pages giving 7,200 date conversions which he was able to work out swiftly with his calculator programs. He also wrote 6 calendar conversion programs for the Egyptian and Alexandrian as well as pre-Julian calendars. His programs were published by PPX-52, a user's library maintained by Texas Instruments for their SR-52 calculator. Few have had such a dramatic need for a programmable calculator, but many have recognized and experienced such a calculator's utility.

A small but rapidly growing international group of calculator enthusiasts has emerged within the past few years. It all started when Richard J. Nelson of Santa Ana, California, got together with a few friends and formed the HP-65 User's Club in 1974. Today the club has a membership of about 2,500 and they are a remarkably diversified group. A member who writes to the club newsletter with a shorter method of generating Fibonacci numbers, or a more efficient way to use flags in conditional testing, is just as likely to be a precocious schoolboy as a college math professor.

Computer hobbyists have also discovered that programmable calculators have a lot in common with their bigger brothers. Sophisticated programming techniques such as vectored processing, linked lists, and self-modifying code, are quite possible on various calculator models. And calculator fanciers have found their favorite pocket programmables very suitable for the inevitable games hobbyists play. There are numerous calculator versions of NIM, Hexapawn, Life, slot machine simulation, and even advanced versions of Star Trek on multiple magnetic cards with enough documentation to fill a small book.

Hewlett-Packard's early lead in the programmable calculator sweepstakes has been aggressively challenged by Texas Instruments in the past few years, and the two companies dominate the field.

But even before the advent of programmable calculators, an unannounced feature was found in the HP-45 scientific calculator. Pressing the CHS (change sign), 7 and 8 keys simultaneously activated a built-in timer. The timer was there because it was a feature of a more expensive calculator in the line, the HP-55. The HP-45 lacked the crystal re quired for accurate timing, so the factory did not provide access to this feature. After users found out how to activate the timer, some even installed their own crystals.

When Texas Instruments introduced the SR-52, searchers after unannounced features had a field day. First it was discovered that program memory and data memory, usually strictly separated on calculators, could be used interchangeably. This led to greatly increased data storage and possibilities for self-modifying code. Another unannounced SR-52 feature was pseudo-code, in which internal code, not representing any calculator key, could be synthesized. Strange calculator behavior and fractured display digits can be produced on the SR-52 by using pseudo-code and direct data storage in registers designed for internal algebraic evaluation.

Two of the latest discoveries are NNN's or non-normalized numbers on the HP-67, and use of a synthesized HIR (for hierarchy internal registers) on the TI-59. The NNNs can be used to create special displays on the HP-67, while HIR operations on the TI-59 make direct storage and register arithmetic possible on memories normally used only by the internal operating system.

Texas Instruments' new TI-59 has some standard features not seen on calculators previously. In conjunction with the PC-100a printer, the TI-59 (and the lower-priced TI-58) support arithmetic printout. They also have tiny, plug-in ROM modules (called CROMs) with 5,000 steps of library routines. Among the sophisticated routines available are a 9x9 matrix determinant and inverse (on the standard Master Library module), Analysis of Variance, and some very inclusive investment analysis programs on the Real Estate/Investment module. Inevitably, TI-59 users are exploring all the ways that CROM subroutines can be called in their own programs, and a book on the Master Library module has just been published.

Hewlett-Packard and Texas Instruments have also begun publishing users library programs in softcover books. One recent publication of special interest is T-I's Programming Aids Pakette, which includes an 8080 Disassembler program, EBCDIC and ASCII code converters, and other goodies.

If you've been ignoring calculators, better take another look at the new programmables. Unlike microcomputers, calculators are single-purpose machines designed for numeric evaluation and possessing only the most primitive data processing capabilities. They are binary-coded decimal machines with a short (usually 8-bit) word length for program instructions, but a greatly extended (56 to 64-bit) word for data storage. Calculators operate a few orders of magnitude more slowly than computers, but you may never notice because the extensive library of "pre-programmed" functions in ROM memory make them seem much faster than they really are.

Programmable calculators are not really designed to compete with microcomputers, but to be pocketable problem-solvers which can be used for anything from navigating a sailboat to teaching arithmetic to schoolchildren. Whatever your reason for buying one, you're likely to discover new uses soon after purchase. After that you can learn the lingo and look for new calculator features you can discover.□

Video Game Techniques In North Star BASIC for the SOL VDM-1

By Robert C.A. Goff, M.D.

INTRODUCTION

Writing a program in BASIC for real-time video graphics or real-time video games is a simple process if you know the inner secrets of your video generator. This article is written for the beginning computer user or for the pro who has never explored the real-time capabilities of BASIC. Although I will discuss the appropriate instructions in North Star BASIC with specific application to the Processor Technology VDM-1, these techniques may easily be applied to any memory-mapped video display, and will run in most versions of BASIC with only minor changes. A partial list of memory-mapped character generators includes the VDM-1 from Processor Technology, PolyMorphic's VTI-64, Solid State Music VB1, and IMSAI VIO.

THE "FILL" INSTRUCTION (POKE)

The FILL instruction (or POKE in other versions of BASIC) is used to place a single character into a specified byte of memory. With a memory-mapped video display, FILL may be used to place a character into a specified position on the CRT. The only difficulty lies in knowing where the character generator resides in your memory. Once you know this you may specify what character you would like in what position on the screen. The VDM-1 in the SOL/20 is usually located at the memory address CC00H, which is 52224 decimal. This means that if you FILL 52224 with an asterisk, the first position on the screen (upper left hand corner) will now contain and display an asterisk. The other positions on the screen have addresses which follow in sequence so that 52224 + 63 is the end of the character of the top line (for 64 character line). Similarly, 52224 + 64 is the first character on the second line, and 52224 + (64*6) is the first character of the seventh line, with the last address on the screen being 52224 + (64*15) + 63 or 53247 for a total of 1024 characters (64 x 14). The usual starting addresses for other systems may be found in the operating manual. The IMSAI VIO is usually at 0000H (0 decimal); the PolyMorphics is usually at 8800H (10048 decimal).

The other necessary information is found on a list of the characters available on your character generator. Since these must be expressed in decimal form, the HEX codes which are commonly found on ASCII character charts must be converted to decimal. This has been done in Table 1 for the 6574 character generator which is standard on the SOL/20. With the exception of the control characters (decimal 1 through 31, and 127) these decimal codes do not vary. It should also be noted that if 128 is added to the value of any character, the character generator will produce the negative image of that character. If the display is set to usually display in the negative image (black on white), adding 128 will reverse any character to its positive image.

The FILL instruction is written in the format:

FILL address,character

and looks like this for placing an asterisk in the first

0	□	NULL	64	@
1	Γ	START OF HEADING	65	A
2	⊥	START OF TEXT	66	B
3	┘	END OF TEXT	67	C
4	≀	END OF TRANSMISSION	68	D
5	⌛	ENQUIRY	69	E
6	✓	ACKNOWLEDGE	70	F
7	Ω	BELL	71	G
8	↰	BACKSPACE	72	H
9	→	HORIZONTAL TAB	73	I
10	≡	LINE-FEED	74	J
11	↓	VERTICAL TAB	75	K
12	⇣	FORM-FEED	76	L
13	←	CARRIAGE RETURN	77	M
14	⊗	SHIFT OUT	78	N
15	⊙	SHIFT IN	79	O
16	⊟	DATA LINK ESCAPE	80	P
17	⓪	DEVICE CONTROL 1	81	Q
18	⓪	DEVICE CONTROL 2	82	R
19	⓪	DEVICE CONTROL 3	83	S
20	⓪	DEVICE CONTROL 4	84	T
21	✗	NEGATIVE ACKNOWLEDGE	85	U
22	⎍	SYNCHRONOUS IDLE	86	V
23	┤	END OF TRANSMISSION BLOCK	87	W
24	⧗	CANCEL	88	X
25	⫲	END OF MEDIUM	89	Y
26	ʕ	SUBSTITUTE	90	Z
27	θ	ESCAPE	91	[
28	◰	FILE SEPARATOR	92	\
29	◱	GROUP SEPARATOR	93	]
30	◲	RECORD SEPARATOR	94	^
31	◳	UNIT SEPARATOR	95	_
32		SPACE	96	`
33	!		97	a
34	"		98	b
35	#		99	c
36	$		100	d
37	%		101	e
38	&		102	f
39	'		103	g
40	(		104	h
41	)		105	i
42	*		106	j
43	+		107	k
44	,		108	l
45	-		109	m
46	.		110	n
47	/		111	o
48	0		112	p
49	1		113	q
50	2		114	r
51	3		115	s
52	4		116	t
53	5		117	u
54	6		118	v
55	7		119	w
56	8		120	x
57	9		121	y
58	:		122	z
59	;		123	{
60	<		124	¦
61	=		125	}
62	>		126	~
63	?		127	DELETE

Table 1. Decimal ASCII Codes.

position, upper left on the video of the SOL:

```
10 FILL 52223,42
```

To make the character appear to move across the screen, it is necessary to FILL addresses on the screen sequentially, and as the character is moved from one position to the next, it must be erased by replacing it with a "blank" character. Such a program would look like this:

```
10 FILL 52224,42 \ REM START ASTERISK IN UPPER LEFT
20 FILL X=52224 TO 52224+63
30 FILL X,32 \ REM ERASE WITH BLANK
40 FILL X+1,42 \ REM MOVE ASTERISK TO RIGHT
50 NEXT X \ REM RETURN FOR NEXT MOVE
```

Moving it vertically is similar, with the exception that the address (X in the above examples) must be incremented by 64 since in a 64 character line two vertically juxtaposed characters differ in their addresses by 64.

```
10 FILL 52224,42 \ REM START ASTERISK IN UPPER LEFT
20 FOR X=52224 TO 52224+(64*15) STEP 64
30 FILL X,32 \ REM ERASE WITH BLANK
40 FILL X+64,42 \ REM MOVE ASTERISK ONE SPACE DOWN
50 NEXT X \ REM RETURN FOR NEXT MOVE
```

With a little practice, it should be possible for you to cause any character or number of characters to move in a specified direction on the screen.

REAL-TIME CONTROL

Movement of a character on the video screen may be modified during the run of a program by setting a variable equal to the last character typed on the keyboard. This is done in North Star BASIC by using the instruction INP(x). The expression in parentheses must be the decimal address of the keyboard (or other device from which the interactive input will be read). On the SOL/20 this is 252, so the instruction appears as:

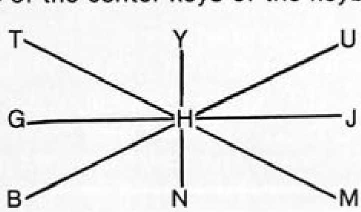
```
10 A=INP(252)
```

This will set the variable 'A' equal to the decimal value of the last key typed on the keyboard. Of course, the last key typed after the command, RUN, is the RETURN key which is decimal 13, so this must be taken into account in the program. To demonstrate the use of the INP instruction, run the following program which will simply print the decimal value of any key (or combination of CONTROL and a character key):

```
10 A=INP(252)
20 PRINT A,
30 GOTO 10
```

The program is continuous, but each time that you touch a key, the value of 'A' in line 10 will change to the decimal value of that key.

Now in order to use these decimal values of various keys to alter the motion of a character on the video screen, you must set up a series of IF statements that will recognize certain keys and assign to a second variable whatever value is needed to bring about the desired movement. For instance, in the game CHASE written by J.J. Sanger and published in the October 1977 issue of INTERFACE AGE, the motion of a "man" is controlled with the use of the center keys of the keyboard:



The numbers are set to a numerical value which, when added to a video screen address, will move a character in the direction indicated by the diagram. Depressing 'H' will cause the character to stand still.

```
10 C=52224+(64*7)+32\REM *** START IN THE CENTER
20 PRINT CHR$(11) \ REM *** CLEAR SCREEN
30 FILL 52224+64,32\REM *** SUPRESS CURSOR WITH BLANK
40 A=INP(252) \ REM *** READ INPUT FROM KEYBOARD
50 B=0 \REM *** B IS THE MOVEMENT INCREMENT VARIABLE
60 IF A=13 THEN B=0 \REM *** NO MOVE AFTER 'RETURN'
70 IF A=72 THEN B=0 \REM *** H=NO MOVE
80 IF A=84 THEN B=-65 \REM *** T=MOVE UP LEFT
90 IF A=89 THEN B=-64 \REM *** Y=MOVE UP
100 IF A=85 THEN B=-63 \REM *** U=MOVE UP RIGHT
110 IF A=74 THEN B=1 \REM *** J=MOVE RIGHT
120 IF A=77 THEN B=65 \REM *** M=MOVE DOWN RIGHT
130 IF A=78 THEN B=64 \REM *** N=MOVE DOWN
140 IF A=66 THEN B=63 \REM *** B=MOVE DOWN LEFT
150 IF A=71 THEN B=-1 \REM *** G=MOVE LEFT
160 IF A=127 THEN GOTO 240 \REM *** DELETE=END GAME
170 REM *** SET BOUNDARIES OF SCREEN
180 IF (C+B)<52224 THEN B=128+B\REM BOUNCE OFF TOP
190 IF (C+B)>53247 THEN B=-128+B
192 REM *** BOUNCE OFF BOTTOM
200 FILL C,32\REM *** ERASE OLD CHARACTER
210 C=C+B \REM *** CALCULATE NEW POSITION
220 FILL C,42 \REM *** PLACE ASTERISK IN NEW POSITION
230 GOTO 40 \REM *** RETURN FOR NEW INPUT
240 END
```

We now have a relatively simple program which will utilize the keyboard for real-time manipulation of character movement over the video screen. And with these elementary techniques (and a little arithmetic), it is possible to use any key, keys, or sequence of keys to bring about any continuous or discontinuous motion on the screen.

THE "EXAM" INSTRUCTION (PEEK)

The "EXAM" instruction (or PEEK in other BASICs) is used to read the contents of any specified byte of memory. In the case of the video addresses, EXAM will tell the computer what character resides in the specified location on the screen. If, for example, your program has two objects (say a man and a shark) moving about on the screen, it would be very informative to learn if the man and shark are in contact with each other. This can be determined by using EXAM to examine the contents of the next space into which each object is about to move. If that next space contains the opposite character, then we can write the program to respond with the appropriate result by using a GOSUB to the subroutine containing instructions for such an occasion. The EXAM instruction uses the format:

```
10 D=EXAM(52224)
```

and will set the variable 'D' equal to the decimal value of the character at the address enclosed within the parentheses. As a demonstration of the functioning of EXAM, try this program which will display each of the 256 characters (positive and negative images) generated by the 6574 character generator in the center screen position

and print (by EXAMining that screen address) the decimal value of the character being displayed.

```
10 LET A=0 \REM *** START WITH CHARACTER 0
20 PRINT CHR$(11) \REM *** CLEAR SCREEN
30 FILL 52224+32+(64*7),A \REM ** 'A' TO CENTER
40 B=EXAM(52224+32+(64*7)) \REM ** B=VALUE AT ADDRESS
50 PRINT B \REM *** PRINT DECIMAL VALUE OF CHR. A
60 IF B=A THEN A=A+1 \REM *** IF THE 'EXAM' HAS
70 REM *** CORRECTLY READ THE CONTENTS OF THE ADDRESS
80 REM *** THEN GO ON TO NEXT A
90 FOR X=1 TO 500\NEXT\REM *** TIMING LOOP
100 IF A=256 THEN 120 \REM *** STOP WHEN FINISHED
110 GOTO 20 \REM *** RETURN FOR NEXT CHARACTER
120 END
```

RANDOM MOTION

In constructing games, it is often desirable to have some character moving randomly about the screen. This can be done by using the same collection of IF statements shown earlier for use with keyboard input. However, instead of getting the movement selection variables equal to keyboard values, a random number generator is used to generate the same 9 choices. This will require the INTeger function to insure that only whole numbers from 1 to 9 are generated:

10 A = INT((RND(0)*8) + 1)

SCORE KEEPING

Scores may be kept by incrementing score variables each time an event occurs which influences the score. They may be continuously displayed by using a PRINT instruction immediately following a screen clear. The screen is cleared by issuing the statement:

10 PRINT CHR$(11)

which is the decimal value (11) of the ASCII code for "vertical tab". By clearing the screen when possible, the PRINT of the score will appear at the top of the screen. You should remember that the screen clear is essential to video games on the VDM-1 since the display will scroll automatically with each RETURN and will change the location on the screen of the various addresses. The screen clear (PRINT CHR$(11)) will reset the VDM-1 latch, so that 52224 is the address of the first character on the top row. An alternative way of displaying scores without clearing the screen is to use the FILL instruction to write the new score characters in the desired location on the screen. Placing text, such as:

"S C O R E   CHICKEN 2   ROAD 4"

by using the FILL, can be quite tedious, but the *result is* usually more aesthetically pleasing than with the PRINT method.

MULTIPLE-CHARACTER OBJECTS

With a great deal of patience, it is possible to program a moving object which is composed of two or more characters. The math becomes tricky if the composite object is to move without coming unglued, so to speak, but such tactics, when successful, are fun to watch. One application that I have found for composite objects is in a game which I wrote a short time ago. The game, soon to be placed in the public domain, centers about a chicken who tries to cross a 4-lane super-highway during rush hour traffic. The object, of course, is to get the chicken across the road without being run over by a truck. The chicken moves continuously, but his direction may be influenced by input from the keyboard. The program, written in North Star BASIC, generates trucks with wheels that appear to roll (by using characters 17,18,19, and 20 in succession).

SUMMARY

Challenging and enjoyable real-time video games may be written in BASIC by using the FILL (POKE), EXAM (PEEK), and INP instructions, on any system using a memory-mapped video display. To do this, you must know the decimal addresses of your video display, and your keyboard.□

Microcomputers in the Home

By Terry Benson

Microcomputers, as you may be aware, have made it possible to own and operate a computer in your home. This interest has been concerned primarily with general purpose computers-those which contain specific "canned" software packages such as monthly budgets, checkbook balancing, etc., and are user programmable. In addition to these "data processing" applications, there is one other area adaptable to the uses of microcomputers: dedicated controllers.

GENERAL PURPOSE VS DEDICATED

A general purpose computer is a user programmable digital computer that may be applied to a variety of tasks by modifying the program.

Most of the microprocessors available today are truly general purpose and, when combined with memory and input/output devices, contribute to a general purpose microcomputer. The distribution of memory in a microcomputer system will usually distinguish the category to which a system belongs. For a microcomputer to be general purpose it must consist primarily of a read/write memory — RAM (Random Access Memory).

A microcomputer system that is composed primarily of Read Only Memory (ROM) or Programmable Read Only Memory (PROM), would be considered dedicated because it continues to operate on just one program. By replacing the ROMs, the microcomputer may be modified to perform a different set of tasks.

ROMs are generally used in these dedicated applications because they provide a much greater density (2048 x 8 bits) than RAMs and are non-volatile (when power comes on, the program is ready to operate). ROMs are also less expensive than RAMs because they are "programmed" or masked at the time they are manufactured. In some cases, the ROM is not desirable since there is an initial masking charge and it is not modifiable; therefore, the EPROM (ultra-violet light erasable PROM) provides benefits of both the RAM (field programmable) and the ROM (non-volatile and dense). PROMs are used in some dedicated applications that must be customized for each particular situation and when implemented in fewer than 100 units. There are ROMs available that are pin-compatible with the EPROMs and thus provide an easy upgrade to ROMs for higher volume applications.

Controller applications are popular dedicated microcomputer applications and include numerically controlled machines, telephone traffic monitoring, signal controllers, sewing machines, and so on.

HOBBYIST COMPUTER SYSTEM

It is estimated that today there are over 25,000 hobby computers. The support for the home computer is provided by more than 50 companies, which manufacture the CPU, memories, peripherals, and/or software. There are also hundreds of computer stores that support the hobbyist.

Many publications are available to help the microcomputer user. The most popular are *Byte Magazine, INTERFACE AGE* and *Kilobaud,* all having to do with the hobby computer industry. These magazines provide hardware and software information, and they provide the basics on how to implement a home computer system built around a microprocessor. In general, these magazines and others, along with many books, are available to help educate the home computerist.

Probably the most help is needed in the area of software — the program language for the computer. Most of the programs available for the hobby computer system are written in BASIC because it is a fairly easy language to learn. But due to the requirement for more memory and its associated higher cost for BASIC, many users will program in assembly or machine language. Assembly language is a little harder to learn than BASIC, but it will allow the programmer to implement some things that cannot easily be done in BASIC, such as control functions. On the other hand, BASIC is more adaptable for data processing and in cases where program size and execution speed are not of concern.

Each BASIC statement is converted into several machine functions but may not be translated as efficiently as if originally written in assembly language. In fact, since most BASIC is translated by means of an interpreter, the time to execute a particular function may take as much as ten times longer to execute than an equivalent assembly language statement. The big advantage of BASIC is that it is easy to use.

COMPUTER APPLICATIONS

Many programs might be used in the home: menu planning, checkbook balancing, weather prediction, and diet planning. In addition to these "data processing" applications, many home computers might be used in controlling appliances, environmental control (heating, air conditioning, solar power), security system (fire and burglar alarms), light control (turn lights on or off anywhere in the house from your computer system).

There are other areas in the home that may be of some benefit not only to you, but to your children. The home computer will have a significant affect on education in the future. Computer programming is one of the obvious subjects that a computer can help to teach. Math is another one that fits logically into a computer architecture. Some others that you might not consider so obvious are history, spelling and English.

Some of the more productive applications that are implemented on a computer system — not necessarily in a home — are in the area of small business accounting systems. Some of the small businesses, a doctor's office or a real estate office, might now consider a small computer system to handle their payrolls or inventory.

There are a variety of applications that the home computer can be used for that we would normally consider business applications.

EDUCATING THE USER

How are all of these people — in many cases inexperienced people — going to learn how to use the microcomputer? Many of the users educate themselves. There are numerous books, several magazines, and an abundance of training classes to assist those learning about microcomputers. Education is often available through the manufacturer (Intel offers week-long courses that help shorten the learning curve for the novice microcomputer user). Many of the computer hobby stores offer educational packages and, of course, most universities are offering credit courses in the theory and use of microcomputers.

SUMMARY

Looking one decade back, the popular computer was a minicomputer and it was difficult then to find a minicomputer at less than $10,000. Now the microcomputer is available at less than 10% of that cost. What'll happen in the next decade? As more applications are implemented with microcomputers, and as microcomputers become available at lower cost, every child will have to — maybe even want to — learn computer programming.☐

The Computation of Direction

By Gene Szymanski

In the daily pursuit of our affairs, we do not find it necessary to have a knowledge of absolute direction, for we are able to find our way about through a recognition of familiar sights and sounds. Even when it is necessary to travel beyond the conventional routes, there are available a multitude of guides to help us reach our destination.

On those rare occasions when we find that we are "lost", the feeling of disorientation quickly subsides once a familiar landmark comes into view, for then we quickly recover our sense of direction.

For our purposes, then, direction is thought of as it relates to some recognized object or prominent feature, such as a structure, highway intersection, or the skyline of a city. Sometimes we find it convenient to extend the scope of our reference by descriptives, such as "to the north, or east", and so on.

The surveyor, navigator, and astronomer require a more precise definition of direction in their work. They are concerned with the measurement of exact positions, often separated by great distances. For their purposes, the concept of direction is of fundamental importance.

THE MEASUREMENT OF DIRECTION

Direction is the angular difference measured in degrees from a reference. For most purposes, we are interested in "true direction" whose reference is the geographic north pole. True direction is measured as an angle whose initial value is 0 degrees at north and which increases in a clockwise direction to 360 degrees. In order to measure true direction, then, it is first necessary to accurately determine the direction of the earth's geographic poles.

For centuries, the magnetic compass has served as the principal instrument for providing a knowledge of direction. Unfortunately, the compass indication of true north is subject to considerable error. The directive force on the compass needle is the result of two forces, one exerted by the earth's magnetic field and the second exerted by iron or steel which may be found in the vicinity of the compass.

The earth's magnetic field is irregular; furthermore, the position of the magnetic and geogrpahic poles do not coincide. This gives rise to error called variation in the direction of north indicated by the compass needle. The amount of variation depends upon location and can be found by consulting a map or chart of the locality. A chart of Long Island Sound, for example, would show that the variation is 13 degrees west. This means that the compass needle is deflected 13 degrees to the west of true north so that a value of 13 degrees must be subtracted from the compass reading to obtain the true direction.

Obviously, it is a simple matter to cope with variation. All that is necessary is to determine its value from the chart, then apply it to the compass reading by addition or subtraction.

The second source of compass error, caused by the presence of iron and steel near the compass, is far more troublesome. The result of this type of error, called "deviation", must be carefully measured for each compass installation before the instrument can be used with confidence. Even then, after deviation errors have been measured and recorded for reference, they are subject to gradual change as the vehicle or ship in which the compass is installed is moved to other locations. Obviously, if the compass is to serve as a reliable instrument, its errors must be known under all conditions of use.

AZIMUTH OBSERVATIONS

Because of the regularity in which celestial bodies appear to move overhead, we are able to observe their positions in the sky and, from this, determine direction. In practicing this technique, we are said to be performing

an "azimuth observation", following a procedure which is used throughout the world to establish direction.

The azimuth of a celestial body is simply its direction from the observer and is measured as a horizontal angle from the north clockwise to 360 degrees. In facing the body, we are also facing the point on the earth's surface directly beneath the body. This point is called the geographic position or "GP" of the celestial object, and, in a strict sense, the term "azimuth" refers to the direction of the GP.

If both the position of the observer and the GP of the celestial body are known, the azimuth of the body can be computed. An accurate direction is thus established which serves as an absolute reference for determining any other direction quickly and simply. Azimuth observations enable us to survey the wilderness, align launching pads in the desert, and determine the error of the ship's compass at sea.

In practice, one sights a celestial body, preferably when it is low in the sky, using a suitable pointer. The pointer is then locked into position, and the exact time is recorded. A celestial "timetable" or almanac is then entered with the time and date of the observation to extract the geographic position of the observed object. Combining this with the position of the observation, the azimuth is computed. This azimuth is the exact direction in which the locked pointer is oriented. The direction of any other object is then found by measuring its angular displacement horizontally from the reference pointer.

METHODS FOR COMPUTING AZIMUTH

The computation of azimuth requires the application of spherical trigonometry. This is because we are dealing with a geometric figure lying on the earth's curved surface (see Figure 1). This figure is a triangle formed by connecting the geographic positions of the celestial body, the observer, and the north (or south) pole.

Although the equations for the computation of azimuth are well established, the solution is tedious and prone to human error. For this reason, many methods have been devised which attempt to ease the burden of computation. These range from tables of logarithms to volumes of "pre-computed" solutions, to which the user must apply a liberal amount of interpolation before arriving at the final result.

In dealing with logarithmic solutions, for example, it is necessary to perform the addition and subtraction of at least a dozen 6-digit numbers after they have been extracted from logarithmic tables. In addition, the "labels" of various angles must be examined during intermediate steps in order to determine how the arithmetic is to proceed.

The solution for azimuth is an ideal application for the small computer, and in this role it replaces pages of mathematical tables. The program can be designed to perform a multitude of preliminary calculations which are necessary to establish the known parts of the spherical triangle. The solution of the trigonometric equations then proceeds rapidly.

The azimuth program shown here is written in MITS 8K BASIC. The entering arguments consist of the observer's latitude and longitude, the Greenwich hour angle and declination of the observed celestial body. Greenwich hour angle (GHA) and declination (DEC) are the astronomical counterparts to longitude and latitude, respectively, and define the GP of the celestial body.

Both GHA and DEC for all the prominent celestial bodies are obtained from the "Nautical Almanac", a publication prepared by the U.S. Naval Observatory and issued by the Government Printing Office and its agents. Because of the earth's motion, the GP of every

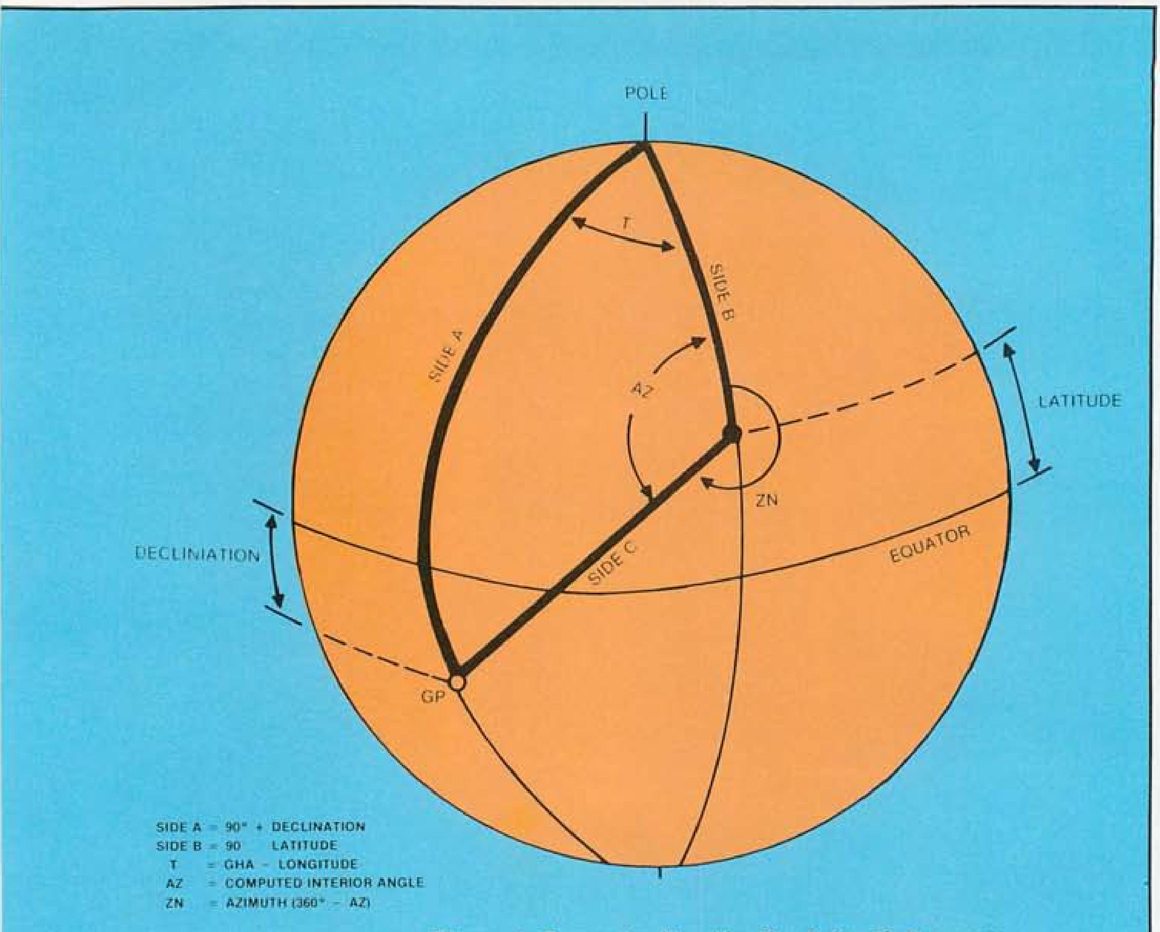


Figure 1. Geometry Used by the Azimuth Program

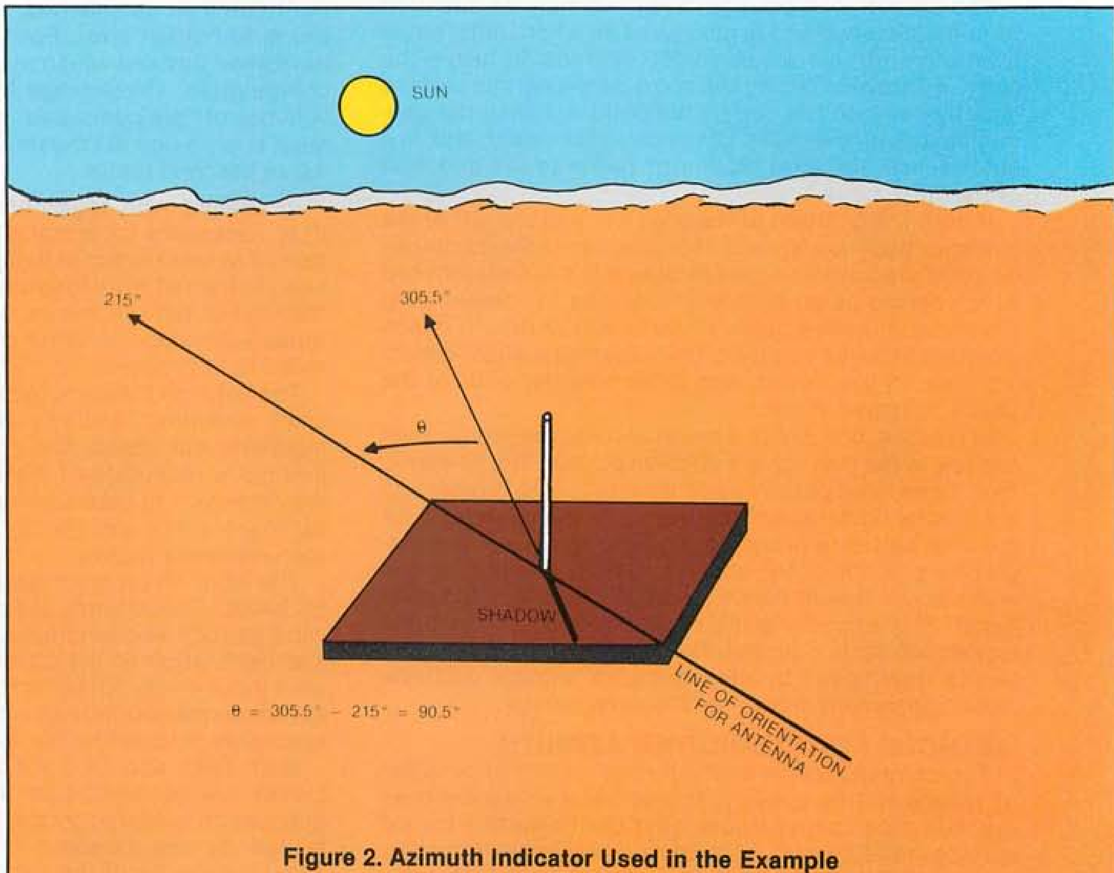


Figure 2. Azimuth Indicator Used in the Example

celestial body is constantly changing so the Nautical Almanac must be entered with the exact date and time (to the nearest minute) of the observation.

The azimuth program solves the spherical triangle (Figure 1) for the angle AZ which is formed by the great circles connecting the observer, the GP, and the nearest geographic pole. Because of the apparent motion of point GP, this triangle expands, then contracts as the celestial body sets in the west, and eventually rises in the east.

The program first processes longitude and GHA to determine angle T. The latitude and declination are then examined to establish two sides of the triangle; the interior angle AZ can then be computed.

Finally, the program converts AZ to angle ZN, the azimuth. This is of great advantage when done by the program since one of four different conversion rules must be selected. This is because the triangle may be referenced to either the north or south pole, depending upon which is nearest to the observer.

The technique of celestial observation for azimuth. . .is performed daily at sea in order to detect any changes in magnetic compass deviations and to verify the accuracy of the ship's gyrocompass.

When started, the program prompts the user to enter input data in the following format: first degrees, next minutes, then the "label". Degrees are always entered as an integer, while minutes are to be entered to the nearest tenth. "Label" refers to the characteristic suffix east or west (for longitude) and north or south (for latitude and declination); there is no label for Greenwich hour angle, GHA.

A longitude, for example, whose value is 134 degrees, 15.7 minutes west would be input to the program in the following format:

(degrees)	134
(minutes)	15.7
(label)	W

At its conclusion, the program prints out the value of ZN. This is the azimuth, or true direction of the celestial body from the observer, valid for the instant of the observation.

MATHEMATICS USED FOR SOLUTION

First the input values, specified in degrees and fractional minutes of ARC, are converted into a decimal degrees format (fractional degrees).

"Local hour angle" (LHA) is now introduced as a parameter to expedite the calculation for meridian angle, T. LHA is found by adding east longitudes to and subtracting west longitudes from Greenwich hour angle, GHA.

Coefficient M has an absolute value of unity. Its sign is now made positive if latitude and declination are both north or both south; otherwise, the sign of M is to be negative.

Side C of the spherical triangle (see Figure 1) is computed by the following formula, derived from the Law of Cosines:

SIN C = SIN L*SIN D + M(COS L*COS D*COS T)

The intermediate value S is not computed as:

S = (C + L + 90 − M*D)/2

The "haversine" function of angle AZ is computed as follows:

HAV AZ = SIN(S-L)*SIN(S-C)/(COS C*COS L)

Interior angle AZ can now be found according to the following identity:

COS AZ = 1-2*HAV AZ

Finally, the value of the azimuth ZN is determined by noting the label specified for latitude and the label of angle T. Four different label combinations are possible and angle ZN is derived from angle AZ according to the following rule:

Label of L	Label of T	ZN
N	E	ZN = AZ
N	W	ZN = 360 − AZ
S	E	ZN = 180 − AZ
S	W	ZN = 180 + AZ

Note: Angle T is assigned the label "W" if LHA is less than 180 degrees; otherwise, the label "E" is assigned to angle T.

APPLICATIONS

The technique of celestial observation for azimuth can be employed wherever there is a need to establish direction. Some typical examples are as follows:

1) Orientation of structures, highways, property lines.
2) Surveys and construction of maps.
3) Calibration of the magnetic compass or gyrocompass.
4) Orientation of solar energy receptors, directional antennas, tracking devices, telescopes.
5) Predicting the position on the horizon at which the sun or any other celestial body will appear to rise or set.

At sea, azimuth observations are performed daily in order to detect any changes in magnetic compass deviations and to verify the accuracy of the ship's gyrocompass.

Professional compass adjusters, when calibrating a ship's compass for the first time, also rely on the azimuth technique for a directional reference. Their approach is to first compute the sun's azimuth for periodic

time intervals, and from this draw a curve of azimuth values as a function of time. The ship is then placed on various headings and the sun's azimuth, as measured by the compass, is noted. By comparing the observed value of the azimuth with the precomputed value, the residual deviation error in the compass can be quickly determined.

Surveyors and mapmakers rely on azimuth observations to provide them with the geographic orientation vital in their work. Once they have obtained directional orientation, a baseline can be established whose length and direction are well defined. The baseline may then be used as a datum from which all other points of interest can be established by triangulation.

EXAMPLE

The following example indicates how the azimuth program would be applied to a practical situation:

A directional transmitting antenna is to be oriented such that it points exactly towards a receiving antenna located several hundred miles distant. Magnetic compass readings cannot be relied upon because of the presence of electrical machinery and a large steel cyclone fence surrounding the transmitter site.

As a first step, the required direction is determined. The transmitter and receiver positions are marked on a great circle chart and the line connecting them is found to have a direction of 215 degrees. A suitable azimuth indicator is now set up next to the transmitting antenna. A simple but effective indicator can consist of a flat sheet of cardboard placed on a level surface, pierced by a rigid, vertical pin.

At a convenient hour, when the sun is low in the sky, a mark is made on the cardboard to indicate the position of the shadow cast by the pin, and the exact time is recorded.

The Nautical Almanac is now entered with the date and recorded time of the observation, and the coordinates of the sun are found to be:

GHA = 81 degrees, 40.2 minutes.
DEC = 22 degrees, 03.2 minutes, north.

Surveyors, mapmakers and professional compass adjusters rely on azimuth observations to provide them with the geographic orientation vital in their work.

The chart indicates that the position of the transmitter is:

Longitude = 20 degrees, 40.2 minutes, west.
Latitude = 41 degrees, 00.0 minutes, south.

These values are entered into the computer azimuth program, and the resulting print-out indicates the azimuth to be exactly 305.5 degrees. In other words, this is the direction which the shadow described at the time of the observation.

A line drawn on the cardboard surface from the mark towards the position of the vertical pin, therefore, points in the exact direction of 305.5 degrees. A second line can now be drawn, offset from this reference "pointer" by an angle of 90.5 degrees to the left (305.5 − 215 = 90.5), to indicate the direction for the antenna. □

PROGRAM LISTING

```
5 REM:PROGRAM"AZIMUTH BY CELESTIAL OBSERVATIONS".
6 REM:BY GENE SZYMANSKI, JAN 3,1978
9 CLEAR 100
10 REM:DATA INPUT MODULE
20 PRINT"ENTER LONGITUDE:"
22 INPUT"DEGREES";A(1):INPUT"MINUTES";B(1)
24 INPUT"LABEL(E OR W)";A$
25 PRINT:PRINT
30 PRINT"ENTER LATITUDE:"
32 INPUT"DEGREES";A(2):INPUT"MINUTES";B(2)
34 INPUT"LABEL(N OR S)";B$
35 PRINT:PRINT
40 PRINT"ENTER DECLINATION:"
42 INPUT"DEGREES";A(3):INPUT"MINUTES";B(3)
44 INPUT"LABEL(N OR S)";C$
45 PRINT:PRINT
50 PRINT"ENTER GHA:"
52 INPUT"DEGREES";A(4):INPUT"MINUTES";B(4)
100 REM:CONVERT INPUTS TO DECIMAL DEGREES
110 FORI=1TO4
120 B(I)=A(I)+B(I)/60
130 NEXT I
200 REM:COMPUTE LOCAL HOUR AND MERIDIAN ANGLES & M.
210 IF A$="W"THEN B(1)=-1*B(1)
220 LH=B(4)+B(1)
230 IF LH<180 GOTO 250
240 T=360-LH: TS=1: GOTO 260
250 T=LH: TS=-1
260 IF T>0 GOTO 280
270 T=-1*T:TS=-1*TS
280 T$="E"
290 IF TS<0 THEN T$="W"
291 LET M=-1
292 IF B$=C$ THEN M=1
300 REM:SOLVE FOR COMPUTED ALTITUDE
310 K=57.2958
320 A=SIN(B(2)/K)*SIN(B(3)/K)
321 A1=M*COS(B(2)/K)*COS(B(3)/K)*COS(T/K):A=A+A1
330 HC=(ATN(A/SQR(1-A↑2)))*K
340 HC=ABS(HC)
500 REM:COMPUTE INTERIOR ANGLE AZ
510 S=0.5*(HC+B(2)+90-M*B(3))
520 H1=SIN((S-B(2))/K)*SIN((S-HC)/K)
530 H2=H1/(COS(HC/K)*COS(B(2)/K))
540 H3=1-2*H2
550 H4=ATN(SQR(1-H3↑2)/H3)
560 AZ=K*H4
561 IF AZ<0 THEN AZ=180+AZ
600 REM:COMPUTE ZN
610 LET X$=B$+T$
620 IF X$="NE"THEN ZN=AZ
630 IF X$="NW" THEN ZN=360-AZ
640 IFX$="SE" THEN ZN=180-AZ
650 IF X$="SW" THEN ZN=180+AZ
651 ZN=INT(ZN*10+0.5)/10
652 PRINT:PRINT
660 PRINT "ZN=";ZN;"DEGREES"
670 PRINT "DONE"
680 END
OK

RUN
ENTER LONGITUDE:
DEGREES? 20
MINUTES? 40.2
LABEL(E OR W)? W


ENTER LATITUDE:
DEGREES? 41
MINUTES? 0
LABEL(N OR S)? S


ENTER DECLINATION:
DEGREES? 22
MINUTES? 3.2
LABEL(N OR S)? N


ENTER GHA:
DEGREES? 81
MINUTES? 40.2


ZN= 305.5 DEGREES
DONE

OK
```

The Personal Management Program

By Carl Townsend

Now that you have got your computer going you have probably found yourself with dozens of projects that need to be done. The computer has multiplied your effectiveness, but how can it manage your time and projects?

Why not use the computer itself to manage the projects? The computer can monitor an inventory of all your existing projects, the relative priority and any deadline dates. This little managing program performs a sort each time the projects are listed, sorting the list in priority and date order.

EASY AS A-B-C

Control begins with planning. What are your long term goals? How do you plan to accomplish these? Can you define some short term goals that would be steps to the larger goals?

1. What resources do you need? (people and materials).
2. What education will you need?

You should try to translate the larger blue sky goals to smaller, realizable and specific subgoals. List these subgoals as projects on a sheet of paper without assigning any priorities. List any relevant deadline dates (income taxes, for example, may have to be mailed before the fifteenth of April). Then go over this list and mark an "A" by those that will give you the most value or need most immediate attention. Those next in order should get a "B", and the next a "C". These values are relative based on your goals and the rewards you envision. For more help on this, read Alan Lakein's *How To Get Control Of Your Time And Your Life.*[1] This list will be used as the input to the computer program and should be updated weekly. A sample list is shown in Figure 1.

PROJECT LIST —

MAILOUT PROGRAM —	Build Module — A
	Sort Module — B
	List Module — B
	Extraction Module — B
	Update Module — B
	Documentation — A
Build system for delivery:	New Book — A
	Letters — A
	Next Newsletter — A
	Church Proposal — A
	Business Proposal — A
Read:	Winter's Book — A
	Magazines — A
Software:	Nutrition Program — C

Figure 1. Initial Project List

USING THE PROGRAM

The program as listed runs in the new commercial BASIC with 24K of memory. It can easily be modified for BASIC-E, Microsoft BASIC, or North Star BASIC. The sort is performed on random files on the disk, so only enough memory is needed for two strings at a time. The sort using a PerSci disk and CP/M takes only a few seconds and the disk head will not drop from the disk during the entire sort. The flow diagram is shown in Figure 2. The program is in Program 1.

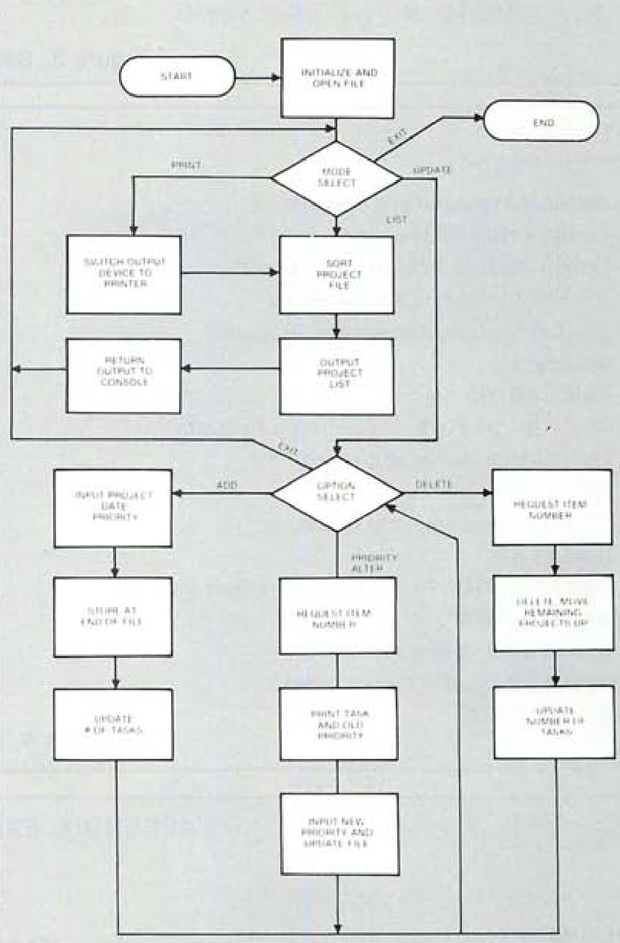


Figure 2.
Personal Management Program Flow Diagram.

GOING TO LARGER PROJECTS

Once the program is mastered, visit your local library and locate books on critical path charting, Ghant charts, and PERT charts. Study up on these and find the methods that seem best for your projects. Use the project codes in this program listing to flag phases of larger projects and you will find this program can monitor progress on your larger projects. Always list the larger project name as well as the name of the particular phase, as:

A 12/20/77 MAILOUT — Build Module
B 12/31/77 MAILOUT — Sort Module
C 12/31/77 MAILOUT — List Module
D 01/15/77 MAILOUT — Extraction Module
E 01/15/77 MAILOUT — Search Module

```
crun task
CRUN VER 1.01
Personal Management Program
Create a New File? n
Option (update,list,print or exit): p
Date: 01/16/78

Personal Task Schedule
Date: 01/16/78
  1  A  01/14/78  Write letters (Center)
  2  A  01/14/78  Next Patterns
  3  A  01/18/78  Continue writing book
  4  A  01/18/78  Document Personal Management Program
  5  A  01/18/78  Mailout — Sort Module
  6  A  01/25/78  Do coordinate maps
  7  A  01/18/78  Repair tape recorder for church
  8  A  01/21/78  Read Winter's Book
  9  A  02/15/78  Church Information System — Proposal
 10  B  01/25/78  Mailout — Update Module
 11  B  01/25/78  Mailout — Extraction Module
 12  B  01/31/78  Read — Corporation Books
 13  B  01/31/78  Read — book on volunteer organizations
 14  B  01/25/78  Business Proposal
 15  B  02/15/78  Assemble 2SIO and Floppy Disk Interface
 16  C  02/28/78  Income Tax — Calculate
 17  C  02/28/78  Nutrition Program
 18  C  02/15/78  Checkout 2SIO and interface
Option (update,list,print or exit): e
```

Figure 3. Sample Project Listing

```
crun task
CRUN VER 1.01
Personal Management Program
Create a New File? n
Option (update, list, print or exit): u
Priority alter,delete,add or exit: a
Job Description: Accounting Program
Priority: B
Date: 01/31/78
19     B  01/31/78   Accounting Program
Priority alter,delete,add or exit: d
Item # : 12
Priority alter,delete,add or exit: p
Item # : 5
Job: A      01/18/78      Mailout — Sort Module
New Priority: B
Priority alter,delete,add or exit: e
Option (update,list,print or exit): l
Date: 01/17/78
Personal Task Schedule
Date: 01/17/78
  1  A  01/14/78  Write letters (Center)
  2  A  01/14/78  Next Patterns
  3  A  01/18/78  Continue writing book
  4  A  01/18/78  Document Personal Management Program
  5  A  01/25/78  Do coordinate maps
  6  A  01/18/78  Repair tape recorder for church
  7  A  01/21/78  Read Winter's Book
  8  A  02/15/78  Church Information System — Proposal
  9  B  01/18/78  Mailout — Sort Module
 10  B  01/25/78  Mailout — Update Module
 11  B  01/25/78  Mailout — Extraction Module
 12  B  01/31/78  Read — book on volunteer organizations
 13  B  01/25/78  Business Proposal
 14  B  02/15/78  Assemble 2SIO and Floppy Disk Interface
 15  B  01/31/78  Accounting Program
 16  C  02/28/78  Income Tax — Calculate
 17  C  02/28/78  Nutrition Program
 18  C  02/15/78  Checkout 2SIO and interface
Option (update,list,print or exit): e
```

Figure 4. Sample of Update

APPENDIX: EXAMPLES OF IDMAS USE

```
READY? create
    *** WHEN FINISHED, TYPE 'DONE' ***
ITEM #1 (KEY)? product
ITEM #2? location
ITEM #3? cost
ITEM #4? maintenance
ITEM #5? type
ITEM #6? life-yrs
ITEM #7? asmb-time
ITEM #8? distribution?
ITEM #9? conversion
ITEM #10? done

READY? add drive-shaft
PRODUCT? drive-shaft
LOCATION? c3-1a
COST? 120
MAINTENANCE? 1
TYPE? carriage
LIFE—YRS? 20
ASMB-TIME? 30
DISTRIBUTION? general
CONVERSION? none

READY? add drive-shaft
  ** 'DRIVE-SHAFT' ALREADY EXISTS **

READY? add unit1
PRODUCT? unit1
LOCATION? 1a
COST? 120
MAINTENANCE? 1
TYPE? carriage
LIFE-YRS? 10
ASMB-TIME? 15
DISTRIBUTION? general
CONVERSION? custom

READY? is there an asmb-time which is less than 30 and
         a location that is in 1a?
** I HAVE FOUND ONE FORM **

READY? display

READY? find a cost which is less than 140*t and an asmb-time
         which is greater than 10*t and a location which isn't 1b.
PRODUCT=DRIVE-SHAFT   COST=120   ASMB-TIME=30   LOCATION=C3-1A
PRODUCT=UNIT1         COST=120   ASMB-TIME=15   LOCATION=1A
** I HAVE FOUND 2 FORMS **
TOTAL COST=240       TOTAL ASMB-TIME=45

READY? end
```

Start the program and the program will request an operation mode:

l — sort and list the current file
p — sort and print the current file
u — update the current file
e — exit

The "l" and "p" mode use the same routine, with the only difference being the output device. Both output the project list (see Figure 3). The exit mode returns the user to the operating system. The update mode, when requested, asks the user for the type of update option desired:

p — alter priority of specified item
a — add an item
d — delete an item
e — exit update mode

The "a" option in the update mode permits the user to add any project to the current list. The project is appended to the end of the current file (see Figure 4). The first record on the file always is incremented by one as each project is added. The file is not sorted until the next list or print mode. As each project is added a project "number" is assigned to the project automatically based on its order in the file. This number is used to alter priority on the project or for deletions.

The "d" option (delete) permits the user to delete any project from the file. The "item number" is requested, and the user inputs the current project number for the project to be deleted. The project is deleted from the file, and all subsequent projects "moved up" to recover the lost space. The first record on the file that indicates the total number of projects is decremented by one. This alters the project numbers for all subsequent projects in the file, and in multiple delete operations the user should start from the bottom of the listing and work up.

The "p" option alters the priority of any project in the file. The current project number is entered and the project priority, date, and name printed. The user enters the new priority. The file is then updated.

The "e" or exit option returns the user from the update mode. No sorts are made until the next list or print mode select.

PROGRAM APPLICATIONS

Notice that the program, as written, does not request the name of the input file with the projects. This is because each person can have their own disk, personal management program, and project file. The management program is stored as TASK, with the project file containing the name of the person who uses the disk. Everybody has their own list of projects, and even the project priorities will vary among family members.

The bubble sort of this program will help you to keep the progress of the project in order. The sort will also keep all phases of a particular program together if they have the same priority and deadline date.☐

PROGRAM LISTING

```
CBASIC COMPILER VER 1.00
Rem Personal Management Program
rem by Carl Townsend
rem last edit date: 1/15/78
        carl.asc$="carl.asc"
        print "Personal Management Program":print
        true = -1
        Input "Create a New File? ";i$
        if left$(i$,1)="y" then goto 80
        open carl.asc$ recl 80 as 1
10      n=1
        if end # 1 then 90
        read # 1,1;q
        input "Option (update,list,print or exit): ";i$
        if left$(i$,1)="l" then goto 21
        if left$(i$,1)="p" then goto 20
        if left$(i$,1)="u" then goto 30
        if left$(i$,1)="e" then goto 90
        goto 10
20 rem print mode
        lprinter
21 rem list mode
        input "Date: ";d$
        print:print "Personal Task Schedule":print
        print "Date: ";d$
        print
        flag = true
        if end # 1 then 25
        while flag = true
                n=2
                flag = false
                read # 1,n;i$
                while q-n
                        read # 1,n+1;j$
                        if left$(i$,1)>left$(j$,1) then
                                k$=i$:i$=j$:j$=k$:flag = true
                        print # 1,n;i$
                        i$=j$
                        n=n+1
                wend
                print # 1,n  ;j$
        wend
25      n=2
        if end # 1 then 10
27      read # 1,n;i$
        i$="   "+i$
        print using "##";n-1;:print i$
        n=n+1
        if (n-1) <> q then goto 27
        console
        goto 10
30 rem update mode
        read # 1,1;q
        input "Priority alter,delete,add or exit: ";i$
        if left$(i$,1)="p" then goto 40
        if left$(i$,1)="d" then goto 50
        if left$(i$,1)="a" then goto 60
        if left$(i$,1)="e" then goto 10
        goto 30
40 rem priority alter option
        input "Item # :";n
        if n>(q-1) then goto 30
        read # 1,n+1;i$
        print "Job: ";i$
        input "New Priority: ";p$
        i$=left$(p$,1)+mid$(i$,2,len(i$)-1)
        print # 1,n+1;i$
        goto 30
50 rem delete option
        input "Item # :";n
        if n>(q-1) then goto 30
        if n=q-1 then print # 1,1;q-1:goto 30
        for s=n+1 to q-1
        read # 1,n+2;i$
        print # 1,n+1;i$
        n=n+1
        next s
        read # 1,1;s
        print # 1,1;s-1
        goto 30
60 rem add option
        input "Job Description: ";j$
        input "Priority: ";p$
        input "Date: ";d$
        i$=left$(p$,1)+"   "+left$(d$,8)+"   "+j$
        q=q+1
        if len(i$)>78 then i$=left$(i$,78)
        print q-1;"   ";i$
        print # 1,q;i$
        print # 1,1;q
        goto 30
80 rem create new file
        create carl.asc$ recl 80 as 1
        n=1:print # 1,1;n
        goto 10
90 rem close files
        close 1
        stop
        end
NO ERRORS DETECTED
```

T.V. Pattern Generator

By Robert Harr, Jr. and Gary F. Poss

INTRODUCTION

The Apple II was chosen with a two-fold objective in mind. The first, to teach BASIC and machine language programming to the authors and second to use it in a variety of social events with an accent on graphic games. The Apple provides excellent overall capabilities, including its portability and color graphics.

After visiting several friends' homes and using many different television receivers as monitors, we noticed that a few receivers did not faithfully reproduce the colors the Apple is capable of generating. The color on these receivers was inaccurate as a result of poor convergence of the three color guns in the picture tube.

We came to the conclusion that a useful alignment program could be written using the Apple's integer basic color graphics. This program would implement the functions of a television bar and dot alignment generator, thus providing a method for converging three gun color picture tubes.

PROGRAM DESCRIPTION

After entering "run", the title page will be displayed (Photo 1). Upon hitting "return", the program menu will be displayed (Photo 2).

In selection No. 1, you are asked to select a solid color (Photo 3). In the event you select a number outside the range of 0 to 15, an error message — Bad Selection, Try Again — is generated. This solid color selection is quite useful in testing for color purity (Photo 4).

Selection No. 2 displays a rainbow of different colors. This selection is useful in adjusting the various receiver color controls (Photo 5).

Selection No. 3 generates a dot matrix which is used to converge the color guns (Photo 6). The "dots" will appear white when alignment is correct.

Selections No. 4 and No. 5 display vertical and horizontal lines (Photos 7 and 8). This selection may be used to adjust the respective gain and linearity controls.

Selection No. 6 produces a crosshatch pattern. Pincushion and barrel distortions are corrected using this selection (Photo 9).

Selections Nos. 4 through 6 may also be used with black and white receivers.

PROGRAM ANALYSIS

Taking a brief look at the flowchart (Figure 1) and the program listing (Program 1), one can quickly determine how the program functions. Statement lines 1 through 80 set up the display portion of the program. This section includes various process, input/output, and decision routines. Statement lines 100 through 630 contain the various pattern subroutines. There is no "end" to the program as various portions are used at different times during an alignment procedure.

CONCLUSION

It is beyond the scope of this article to provide television convergency instructions. However, several excellent books are available on the subject.[1]

A word of caution must be presented at this time. Television receivers use very high voltage (typically 10-35 thousand volts) to operate the picture tube. If you find that your receiver needs alignment and you are not familiar with the HAZARDS and test procedures, we strongly suggest you do not attempt to align your receiver without proper safety precautions and guidance. If you are uncertain of your abilities, you may wish to employ the services of a qualified technician.□

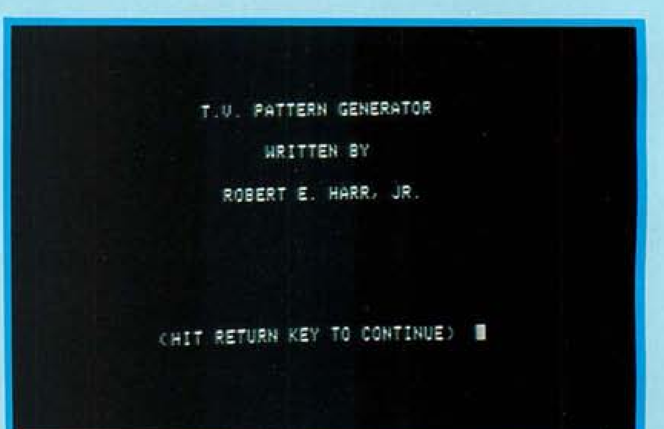


PHOTO 1

REFERENCE

[1]Color TV Servicing, by Walter H. Buchsbaum, publishers: Prentice Hall.

ABOUT THE AUTHORS

Robert Harr, Jr. is Shift Supervisor of Computer Operations for the University of Texas at Austin, Texas.

Gary F. Poss is Senior Electronic Technician with Production Test Equipment Services for IBM at Austin, Texas.

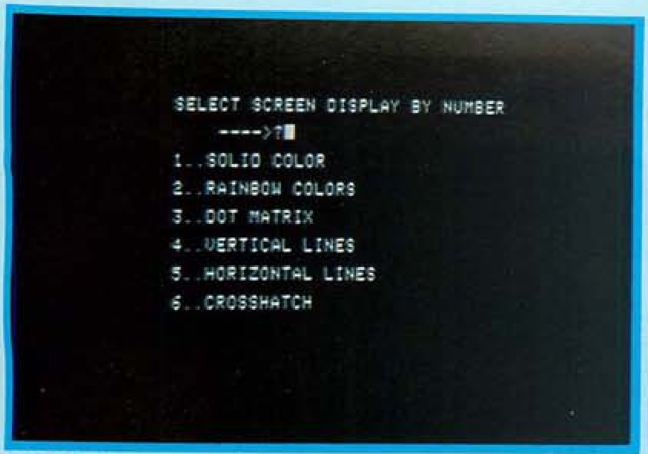


PHOTO 2

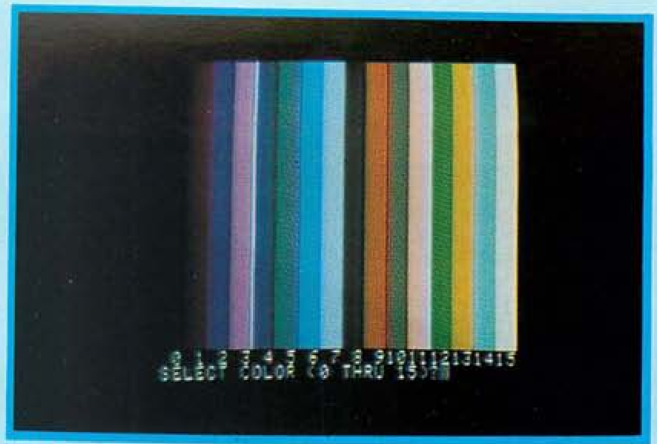


PHOTO 3

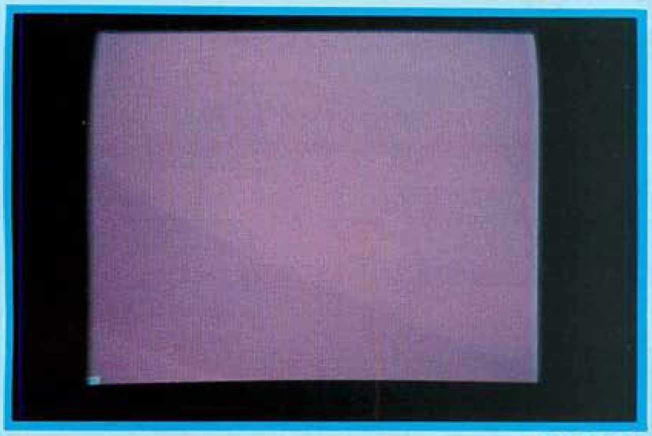


PHOTO 4

PHOTO 5

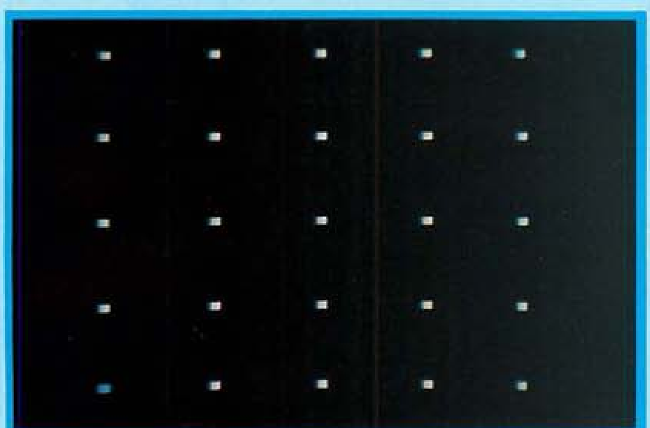


PHOTO 6

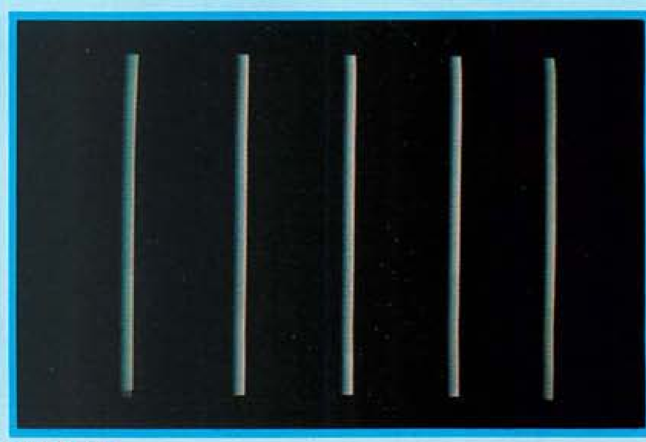


PHOTO 7

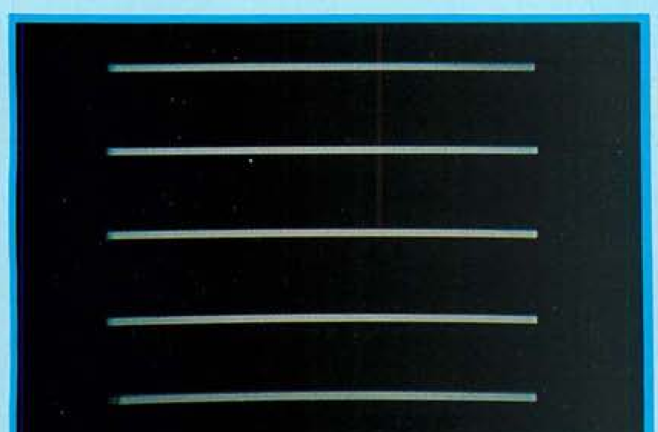


PHOTO 8

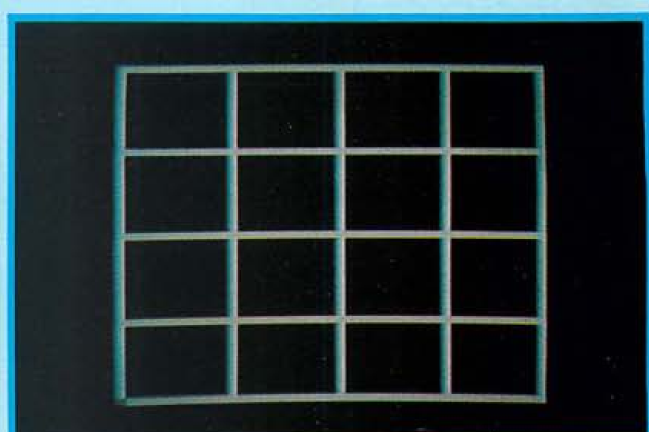
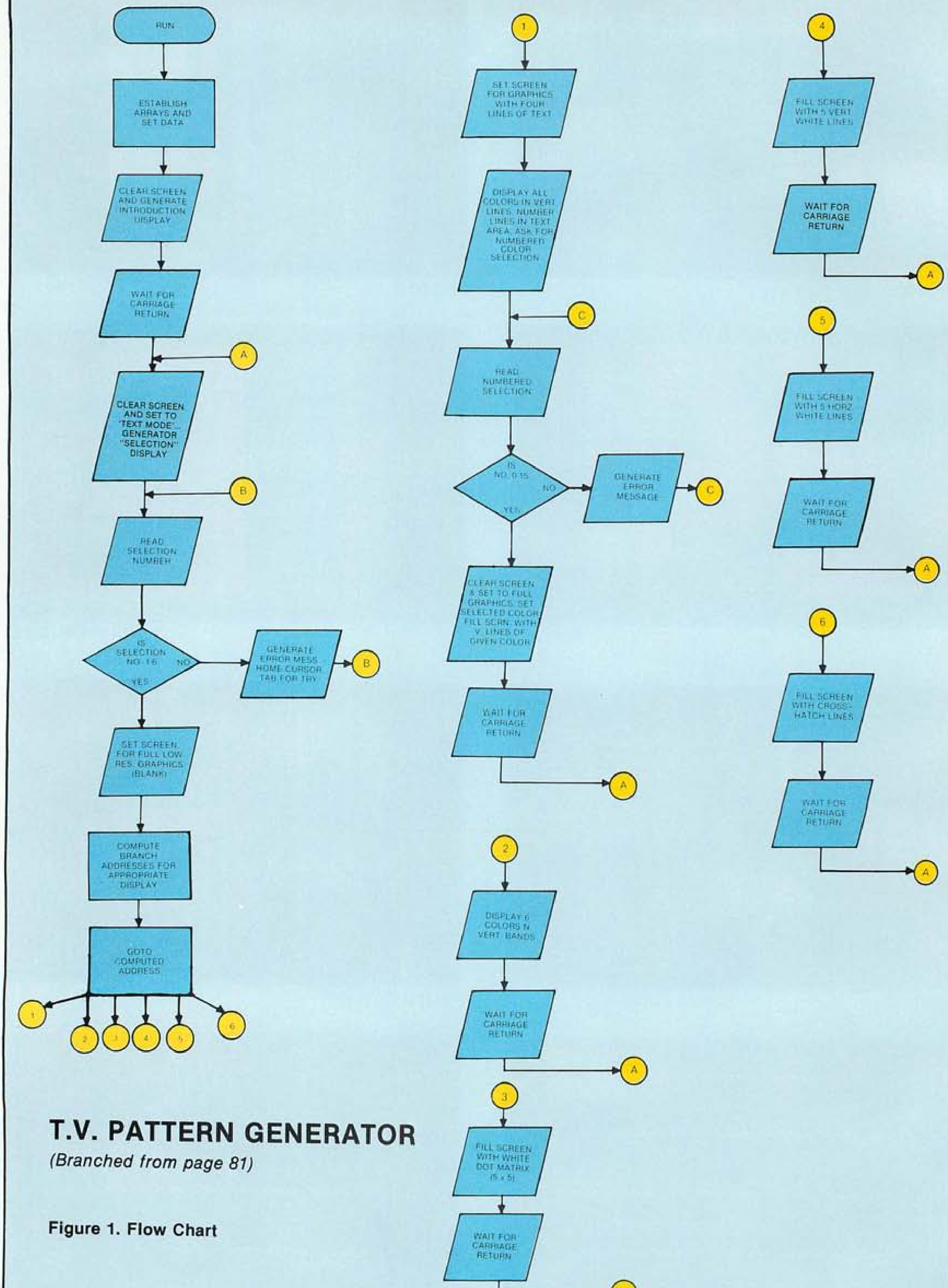


PHOTO 9



T.V. PATTERN GENERATOR

(Branched from page 81)

Figure 1. Flow Chart

Program 1 Vectored to page 160

Considerations for Computer Implementation in A Small Business

Part 4
Selecting the Computer System Software

By Roger Williams

THE SYSTEMS SOFTWARE

The systems software is the set of programs which allows the computer to function as a coherent system, and which is the foundation underlying and required by the applications program. The systems software usually is comprised of two parts, an operating system and a high-level language implemented by an interpreter or compiler. The operating system always is essential, but the language is not necessary in occasional situations where the applications program is supplied directly in machine code.

SELECTING THE OPERATING SYSTEM

The operating system is the heart of the computer, being the program which coordinates all the different parts of the computer, including mainframe, peripherals, the high-level language, and the applications program. Since a disk is always part of the small business system, the operating system also is a disk operating system, or DOS.

The reliability, compatibility, and capabilities of the DOS are the essential selection criteria, of course. Reliability is ensured by selecting a DOS which has been used by enough systems and for a sufficiently long time to detect and eliminate most errors. Also, reliability depends upon large amounts of user-oriented documentation to facilitate accurate and complete usage of the system by the businessman, and to enable him to explore its full capabilities. Programmer-oriented documentation also is essential to permit modifications and adaptations of the DOS to specific configurations of the mainframe and peripherals, and to enable adequate error tracing when necessary.

Next to reliability, compatibility is the most important consideration for the operating system selection — compatibility both to the computer system itself, and to software and hardware available in the industry. Thus the DOS must accommodate the hardware utilized in the system, and also must interface properly to the high-level languages and applications programs. Additionally, it should be multiple-sourced, if possible, and should be used widely by other software such as applications programs.

Each operating system will function properly only with certain hardware configurations, depending upon the CPU, the controller boards, and the specific disk drives. Ideally, the operating system would be purchased as an integral part of the disk drive and controller, unless professional aid is available — and even with a professional, adapting an operating system to an alien hardware configuration can be costly. An example of DOS with such hardware dependency is CP/M, which is designed for the 8080/Z-80 CPU.

Other specific hardware requirements for the DOS affect the controller, I/O boards, and memory. Especially relevant in such hardware are features such as interrupt operation of the controller and I/O boards, intelligence on the controller, and DMA operation or bank-switching on the memory boards. In these cases, even purchasing the disk drive, controller, and DOS as a package is no guarantee of compatibility.

In addition to hardware compatibility, the selected DOS should be compatible to software needed by the business, including languages and applications programs. This means that the DOS should be compatible with a large variety of languages with their associated compilers and interpreters, including BASIC, FORTRAN and possibly PASCAL.

This compatibility is not to be taken for granted, because many high-level languages, or their compilers or interpreters, are designed to work only with a certain DOS, and no others. The compiler or interpreter must not merely work with the DOS, but also must utilize the full power of the DOS, including both data and program accesses, and including a sequential, random, and possibly even indexed sequential accesses — with logical file names or with data-content keys. Examples exist where a particular interpreter or compiler may work with a DOS, but where the programs written in the language are not able to utilize the full powers of the DOS.

The variety of languages available to small-business mini and microcomputers includes BASIC, COBOL, FORTRAN, APL, PLM, PASCAL, FORTH, LISP, FOCAL, and ALGOL, plugs a variety of assembly languages. Such a wide selection of languages for the small computer underlines the need for choosing the DOS, and CPU, which can accommodate at least a good sample of compilers or interpreters, as needed by the business.

Just as many compilers are designed for a specific DOS, many applications programs are designed for a specific compiler, which in turn requires a specific DOS. Other applications programs, if supplied in machine code, likewise may require a specific DOS directly. A similar situation exists for custom program-development software, which almost always presumes a particular DOS. Thus adequate compatibility with the industry requires the selected DOS to be utilized also by a wide variety of applications and program-development software.

Operating systems which are not as widely-used must be evaluated carefully in terms of existing software support and the costs thereof. Such a DOS might be more capable than some widely-used systems, and it might have more software needed by the business, but this is not the usual case. A non-standard operating system can cost heavily in the long term by requiring special adaptations of existing software, or by requiring extra

custom programming. Similarly, adaptations of a widely-used DOS itself to non-standard hardware may prove difficult.

Another point to consider in the selection of the operating system is the capabilities of the DOS. A wide variety of features should be looked for, including efficient usage of disk storage space, powerful file manipulation, and inclusion of major utility programs such as a text editor, assembler, debugging facility, and program writing aids.

The most important function of the DOS is the storage and manipulation of programs and data, preferably with efficient use of the available storage on the disk. One feature to achieve such efficiency is dynamic (during program execution) allocation and release of variable-length files, in those systems which use files rather than the more sophisticated virtual-memory techniques. A related feature is the reorganization of file space and the reclamation of formerly utilized space. All of these are most effective if the user need not intervene.

Other file functions of the DOS include the ability to transfer files to other disks or peripherals, and to restructure files by processes such as merging, concatenating (linking together), or extracting subsets, of files. File directories, and extensive indexing of data are useful and even essential for some applications. File editing, search, and sorting capabilities are similarly important. Batch-processing, whereby one file may be used as a control file to store commands for the DOS to utilize as the program executes, is especially useful to enable the computer to perform a complex series of tasks without interaction on the part of the user.

Any DOS includes a number of inherent capabilities primary to the function of the DOS. It usually includes at least one compiler or interpreter capable of utilizing most of the power of the DOS. Additionally, and of less direct concern to the business user, but important for custom programming efficiency, the DOS may include a variety of major utility programs, such as a text editor, assembler, and debugging package.

A text editor which incorporates string substitutions, searches, and block moves is useful, as also is a text-output processor which provides indentations, titles, left and right justifications, and pagination. Preferably, such a text editor would be character oriented as well as line or block oriented, and should be capable of handling special characters such as those used to write programs.

If any programming is done in assembly or machine language, a good macro-assembler is a *must* to enable a library of parameterized program modules to be defined and then to be conditionally utilized, thereby eliminating the repetitious coding and debugging normally required for identical or similar subroutines. A good macro-assembler also provides for linking module names and variables to those of other modules, and for allowing segmented execution of those assembly programs which, if not segmented, would exceed memory capacity of the mainframe.

Another essential tool for efficient custom programming is a dynamic debugging package, such that assembly programs may be scrutinized both statically and dynamically (during execution) to locate sources of errors. Static examinations include symbol tables, cross references, and examination of memory in convenient forms including ASCII tables and disassembly (conversion) of machine code to assembly language. Dynamic monitoring includes the ability to set breakpoints or conditional breakpoints, at which execution stops and reveals contents of registers and desired memory locations. Dynamic monitoring also includes the ability to single-step through critical areas, with similar register and memory printouts. A capability of tracing the sequence of instructions, complete with all references up to the breakpoint, is especially useful to locate errors.

If programming is done in high-level languages such as BASIC, static and dynamic examination tools similar to those for assembly language are available and useful. The static program-development tools can be entirely independent of the compiler/interpreter, and useful for all languages, provided the programmer is willing to utilize structured and modular programming techniques. At least one supplier has software which can examine and collect such defined modules, then reveal the calling organization and cross-references, complete with formatted printout.

SELECTING THE HIGH-LEVEL LANGUAGE

The computer itself can understand only sequences of ones and zeroes, organized into 8-bit bytes or 16-bit words. Therefore, all programs, in any language, eventually must be converted into these basic units before the program can run, or execute. This conversion is the task of the interpreter or compiler, which is named according to the language it is designed to translate.

The interpreter does the translation during execution of the program, in contrast to the compiler which does such translation prior to execution, so that the actual execution may proceed directly without the time-consuming translation in each step along the way.

In evaluation of the compiler or interpreter, the areas of reliability, compatibility, and capability are important. Reliability factors involve the amount of usage and testing of the compiler together with the availability of requisite documentation. In addition, the syntactical structuring of the language affects the reliability of the programs written in the language.

The selected compiler or interpreter should be in use for enough time, and with enough users, such that most errors have been caught, just as needed with the DOS. An unproven compiler is frustrating and costly, creating errors which can be exceedingly difficult to trace. Additionally, the documentation for both user and programmer should be complete and correct. The user needs to have a clear, thorough, and detailed explanation of what the compiler does in each line of code, and the programmer needs to know details involving error tracing and trapping, I/O options, and machine-language interfacing.

Aside from reliability, compatibility factors of the compiler/interpreter are essential, requiring multiple-sourcing, wide-usage of the language and the specific compiler, and also requiring hardware compatibilities with certain hardware — all similar to compatibility requirements of the DOS.

Any compiler or interpreter selected from the list of those available for the specific hardware and operating system also should be second-sourced, in the event that one version proves to be unusable, or no longer can be supported by the original supplier. This potentially could avoid the necessity of converting an applications program to another compiler, or even to an entirely new language.

The language selected also should be widely-used by a variety of applications programs. Although COBOL is used extensively in large business installations, the most available and commonly-used language in the new small business microcomputers is, by far, BASIC. Nearly all applications programs are written in BASIC. Most versions of BASIC are interpreted, although a few compilers exist and are preferable from the standpoint of speed. Also, increasing amounts of applications programs in the future are likely to be written in compiled BASIC, as opposed to interpreted BASIC, because the compiled code can be provided without the original source code, thereby eliminating or reducing unauthorized tampering and copying of the program, and also reducing memory requirements.

It is important to note that an applications program written in one dialect of a language will not necessarily

run with any other dialect of the language. Such language standardization is practically non-existent except for theoretical discussions at high levels. With BASIC, for example, there are as many versions or dialects as there are manufacturers. These incompatibilities require that the specific dialect of the language be selected as carefully as the language itself or the DOS, and in fact may well determine the DOS and ultimately the CPU.

An applications program which is written in one dialect of a language may be adapted to another dialect, but only if the source code is provided with the applications program, and this seldom is the case, as discussed earlier. Even with source code, although the simpler programs may be converted easily, the complex programs are not easily converted. They utilize sophisticated and extensive disk accesses which are very different between dialects. Other difficulties for translating dialects involve string handling which is sometimes radically different, and I/O protocols and machine-code accesses. The best strategy is to implement the specific compiler or interpreter for which the desired applications programs are available without conversion.

Hardware compatibilities of the compiler or interpreter involve most of the same issues as the DOS. Some compilers may require a specific type of disk controller, especially if using a built-in DOS rather than an external one. They also may require certain memory-switching capabilities or interrupt-handling features on the disk controller or I/O modules. And most important, the compiler or interpreter is always written for a specific CPU. In this respect, the 8080/Z-80 CPU is the most advantageous, having a large variety of both compiled and interpreted dialects of BASIC, at least three dialects of FORTRAN, and at least one version each of COBOL, PASCAL, FORTH, FOCAL, APL, ALGOL, and possibly others. The 16-bit minis also have a proliferation of these languages, usually at higher cost.

The overall characteristic of the language itself is the power of the language, which refers to the amount of calculation or manipulation which can be specified in each line of code in the language. A language which can do in one statement that which would require ten statements in another language is much the more powerful language. Power also refers to the speed of execution of the manipulations, and to the syntactical structuring referred to earlier.

The importance of power is its effect upon the cost of program development and operation. Programming costs, including all documentation and debugging, are about $10 per line of code, whatever the language. The higher power language thus can be highly cost-effective for custom programming specific tasks. Such cost-effectiveness may need to be weighed against possible slower execution of the more powerful language, especially if it is interpreted, and also must be weighed against incompatibilities resulting from a possible obscure language that otherwise might be admirably powerful and capable for the desired tasks.

The specific dialect of a language, expressed by the compiler or interpreter, determines many features which are essential to the small business, and which must be incorporated into the compiler or interpreter selected by the small business. Such specific features comprise four major categories — those of execution capabilities, external reference capabilities, convenience of design and documentation, and ease of coding.

The execution capabilities involve numerical precision, string manipulations, accommodation of special characters, error tracing facilities, debugging support, and whether the language is interpreted or compiled.

Numerical precision should be at least 11 digits, and preferably more, up to 16 digits. An 11-digit accuracy will allow total amounts exceeding 999 million dollars, sufficient for the small business. Lower precision may be adequate for the very smallest of businesses, but consideration must be given to round-off errors, especially in calculating differences of large numbers.

Suitable character-string manipulation includes the ability to extract any desired subset from the string, and for concatenating strings. The most convenient form of string handling allows a single variable to contain a complete string of characters, in contrast to some dialects which allow only a single character for each variable or element of an array.

One important related feature is a special I/O facility to allow the compiler to input or output special characters without influencing operation of the program. Examples would include the ability to input ";", ",", ":", """", or "''" — which can be a problem in some language implementations.

An essential feature is the ability to trap errors during execution without the error causing a return to the operating system and possibly losing data stored in the executing program. An example of such an error would occur when the operator inserts an illegal character (typographical error) while entering data. Without error-trapping, an applications program would be likely to *crash* unless it is written very cleverly.

No truly adequate compiler will be without a dynamic debugging facility, such as that mentioned earlier in the discussion of the DOS. For efficient error location, the compiler at least should enable the setting of conditional breakpoints with the ability to examine and change the contents of all variables. Also needed is the ability to single-step the program line by line with similar access to variables, and then needed is an error tracing system which indicates the sequence of statement numbers leading to the breakpoint.

Also directly affecting execution performance is whether the dialect utilizes a compiler or interpreter. As mentioned earlier, the interpreter translates the lines of code during execution of the program, whereas the compiler translates the code to machine language prior to execution. Such compiler-derived machine code executes at least ten times faster than the interpreter, and does not require the continual residence of a large interpreter in memory. Unfortunately, the compiler does not allow interactive corrections and monitoring of program execution as does the interpreter, although an incremental compiler will do so. Another type of compiler, the pseudo-compiler, is identical to the compiler, except that the machine code generated is that of a hypothetical machine which is also more powerful. This code can be interpreted quickly by the computer during execution to produce its native machine code at speeds approaching those of the true compiler, but with far less program storage than even a native machine-code program would need, due to the power of the hypothetical machine code.

These execution capabilities are closely related to another set of capabilities of the specific compiler or interpreter — those of external references. These include references to memory, selection of I/O channels, disk access for programs and data, access to machine-code subroutines, segmentation of execution, and library creation and utilization.

Memory access involves the ability of the compiler to insert and to retrieve any desired data to and from any memory location, including locations not necessarily within the address occupied by the compiler or program. The most frequent formulation of this ability is in the "PEEK" and "POKE" facility of BASIC.

The selection of I/O channels is useful for a variety of tasks, such as switching printout from the CRT terminal to the printer. The most useful form of I/O channel selection is dynamic switching such that the program itself may cause I/O changes without operator intervention. The use

of logical names or numbers, as opposed to actual, physical I/O ports, is strongly advised, so that different physical devices can be attached to any logical name independently of the program, eliminating the need for modifying the program itself for different I/O needs.

Disk access by the compiler must be reasonably sophisticated, because disk access is the heart of a business system. Both sequential and random files, and preferably also indexed sequential files (using data keys to locate the desired data), must be supported for both data and for programs. The length and location of files should be defined dynamically, as discussed earlier for the DOS. The addressing of random files should be defined by file name or by data-content, but not by direct physical location such as track and sector, which is cumbersome, and which is the responsibility of the DOS rather than of the user of the language.

The program also should be able to access assembly-code or machine-code subroutines, complete with full parameter-passing capability. Such accesses ensure that the applications program can perform many functions impossible to do within the syntax of the language, and enables the user of extremely fast machine code to execute bottlenecks in the applications program.

A crucially important aspect of the external reference capabilities of the compiler, often overlooked, is the ability to execute segment by segment, such that the total program may exceed the available computer memory. With such a facility, each segment causes the loading and execution of the next segment, with all data remaining intact.

And finally in the list of external references, is the capability of having a library of subroutines available on the disk which can be accessed by the program whenever needed. Preferably these would have variables and symbols linked and loaded by the compiler/interpreter, and also would permit using programs in the library which are coded in languages different from the calling program, such as BASIC applications program accessing programs or subroutines written in FORTRAN, in assembly language, or in other languages.

The third set of features of the specific compiler, and important in the small business, involves convenience of design and documentation, which is especially valuable for custom programming development. Such convenience includes considerations of the power of the compiler, the structuring in the syntax, the accommodation of extensive comments, and the ability to use long variable names and both upper and lower case characters in the coding.

Also affecting the ease of design and documentation is the syntactical structuring of the language. One element of syntactical structure supports breaking the program down into small isolated modules which are progressively nested from the general to the detailed levels. Another element of syntax supports structured coding protocols and statements. Finally, the capability of defining and manipulating complete data structures with simple statements is a powerful tool for designing and documenting the program.

An easily overlooked feature to facilitate convenient self-documentation of a program relates to comments and variable names. Good programming practice recommends a generous use of notes and remarks throughout the program, explaining procedures, delimiting modules specifying input and output variables and conditions, and other essentials, to enable another programmer to understand, debug, or modify the program. These notes can occupy enormous amounts of program storage space with most interpreters, but some compilers can accommodate remarks without increasing storage required for operation of the program, by relegating the notes to source-code only.

The use of long variable names, not limited to two to six characters, is useful to describe the meaning of the variables. This can save considerable time and cost for custom programming of large or complex packages. A similar capability is the use of lower-case characters for statements and variable names to enhance readability of the program and variables.

The fourth category of specific compiler features needed for the small business is the ease of coding, as needed for custom programs. Obviously, the power of the compiler and the structure of the language contribute to the ease of actual coding, but certain other factors additionally contribute, including editing features, print formatting, and tabulation features.

A text-editor incorporated into the compiler frequently is necessary, even if the DOS has a text editor, because the DOS editor might reject many of the symbols needed by the compiler. The usefulness of the editor resides in the ability to insert, delete, modify, or move code lines without completely rewriting the line. An especially useful additional feature is the ability to replace specific character sequences, such as variables, with an alternative, without needing to do the replacement individually and manually for a large number of variables.

Print formatting features are especially relevant to the small business for generating invoices, checks, purchase orders, accounting reports, or whatever else is needed. Especially important is control over the exact number of characters and format in dollar amounts, and a large number of compilers will not permit such precise formatting.

Closely related to print formatting is horizontal and vertical tabulation to any desired column or line, and the ability to tabulate to the tops of forms, perhaps automatically after a specified line number is reached.

APPLICATIONS SOFTWARE

Applications programs provide the capability for which the computer ultimately exists. Although the hardware, operating system, and programming language are the foundation upon which the applications program rests, it is the applications program itself which does the work needed, such as accounting, text editing, or other tasks — and also which is identified as the program in the computer.

Applications programs, including custom programming, will represent the largest expenditure on the computer system over the long term, progressing as the computer is utilized for increasing workloads. In contrast, the hardware and system software represent one-time costs rather than continual growth and experimentation.

The applications software is acquired from two basically different sources. The most common and economical source is the applications package designed specifically for certain tasks the businessman requires to be done. Sometimes the available selection of such packages is insufficient to accommodate the needs of the businessman, creating the necessity of the remaining means of acquisition, custom programming. Sometimes an applications program is sufficiently close to doing the tasks that only a modest amount of custom modifications are needed to adapt the package to be suitable. Other times, the applications package is too inflexible or simply is not available, in which case a systems analyst or programmer must be engaged to design and implement a complete, and possibly complex, program or system.

SELECTING THE APPLICATIONS PACKAGE

The selection of applications programs is often tedious and requires study — but there is no substitute

for the businessman himself unless he hires a professional systems analyst — because only he knows what he wants to accomplish and how he wants to do it.

In keeping with the general theme of these articles, the selection is based upon the essential criteria of reliability, compatibility, and capability.

As with all other software, reliability demands that the program be sufficiently widely used, over a long enough time, that most of the bugs have been discovered and corrected. Pioneering new software is not the job of the small businessman. Another requirement for reliable software is that extensive and accurate documentation be provided for both the user and the programmer. Obviously, the user must be instructed about all the details of program operation so that he can use it fully and accurately, and so that errors can be traced if necessary.

Indeed, if extensive user and programmer documentation is not available, it is a warning. A well-written and reliable program would be at least half-documented before being written — a reliable program without documentation is almost a self-contradiction!

Preferably, the documentation should include the source code of the program to facilitate programming special adaptations and modifications such as those needed for specific disk controllers or I/O devices, especially printers. Unfortunately, because of the proliferation of unauthorized copying of applications software, many suppliers will refuse to provide source code, thereby severely limiting the flexibility and even the reliability of the program. A useful compromise would be for the supplier to provide source code for the sections of the program most sensitive to the specific hardware or underlying software, to facilitate custom applications.

A reliability factor closely related to documentation for the applications program is the style by which the program is written. It should be fully structured with modular blocks performing different tasks, and the code should use the structured programming constructs whenever possible. Such a style often indicates the care and coherence of thought of the programmer and analyst — in contrast to sloppy programming without modular design which probably will function similarly, with intricate, sporadic errors and with neither anticipation nor checking of special circumstances. A bonus for modular and structured design is that those modules sensitive to specific hardware and software variations can be isolated, and source code can be provided to the user to create the flexibility discussed above.

One of the most important, and neglected, factors influencing reliability of the applications program is the thoroughness of testing and debugging. The assurances resulting from wide use are helpful, but are not a guarantee. It is prudent for the businessman to question the supplier thoroughly about testing and debugging procedures utilized by the supplier, and to beware if obvious errors arise, for these are indications of inadequate testing procedures. There is no excuse for obvious errors to exist on commercially-available software, such as those errors which arise from simply trying the software for a demonstration.

One procedure the businessman can utilize is to test the software package as thoroughly as possible, dreaming up the situations to enter every possible module with as difficult an example as possible. If the source code is available, then the full debugging and testing procedures specified later in the section on custom programming could be utilized.

In addition to reliability, one essential criterion for the selection of any applications package is that it be compatible with the underlying hardware and software, without requiring any conversion except possibly the usual type of I/O driver patches. The most underestimated type of conversion is converting a package from one dialect of a language to another dialect. This is a job for a professional, especially if disk references are used, as is nearly always the case. All changes created by the conversion must be checked thoroughly to prevent dangerous but subtle errors which might not appear in any immediate or obvious manner, remaining totally undetected or detected too late.

Obviously, the applications program must be compatible with the DOS as well as the compiler or interpreter — and even with the CPU, if no compiler is utilized. Usually these types of incompatibility will cause the program simply not to run, but some incompatibilities are more subtle, such as those which depend upon different detailed specifications of the CPU.

In addition to compatibility with the system software, the applications program must be compatible with the hardware, including both mainframe and peripherals. The applications package may need special modifications to enable proper operation, including those involving memory size and allocation, specific I/O protocols, and peculiarities of the memory and CPU such as timing loops, bank switching, etc. One example of such adjustments is adapting a specific printer, complete with a variety of spacing and tabbing controls, to the applications package. Another example is adapting a particular CRT terminal, complete with cursor controls and protected fields, to the program.

The capabilities of the applications programs are the last criterion of selection, and involve several factors. The most obvious factor is that the package be capable of the needed task. Additionally important is the verification of such suitability through careful study of the operator's manual, and then verification by an actual demonstration of the capabilities. Another capability factor is that the program be easy to use and be oriented to the non-programmer.

The possibility of finding an applications program for any needed task depends upon the CPU and DOS, as discussed before. For the more widely used CPU and DOS, the list of currently-available programs is growing rapidly. The most common applications are accounting programs such as payroll, general ledger, inventory control, accounts receivable and payable, purchase orders, job-costing, and mailing lists. Other applications include task-management with priorities, text-editing and formatting packages, and at least one sophisticated general data-base access and manipulation package. Specialized applications include medical accounting packages, real estate listings, and process-control and analysis for manufacturers. The best way — indeed the only way — to verify such capabilities for any given applications package is to buy and study the manuals, preferably with a demonstration. Buying a package either without manuals and study, or without a demonstration, is risky, and with neither, sight unseen, is asking for trouble.

One extremely useful tactic to ensure the suitability of applications packages is to engage the services of an independent consultant who can evaluate the features of the different packages. Another excellent tactic is to ask the supplier for the names of other users nearby who would be willing to discuss the software. A brief discussion with a satisfied — or unsatisfied — customer rapidly will clarify problems and advantages of the package, in terms of both design and operational reliability. Such a method may elicit opinions in addition to facts, thus it is important to recognize that other users may not have the same needed tasks or evaluative priorities. For example, a special capability may be exactly what one user needs, but exactly what another user might find frustrating and cumbersome to use.

One of the most frequent mistakes in buying applications packages is to assume that the computer will do

certain functions, even though these functions are not explicitly stated in the manuals and never have been demonstrated. The assurances by the supplier are often based upon incomplete understanding of the needs of the businessman or upon his own optimism. An example would be an accounting package where audit trails and invoicing data are not kept in the accounts receivable file, but where the businessman, a bit awed by the apparent capability and storage of the computer, easily could assume otherwise.

The last factor of capability is that the program be easy to use and be oriented to a user who is not a programmer. For example, in an accounting program, the names of suppliers, customers, and brands should be referenced by name, not by number. The computer is supposed to make life easy — not complex, such as by requiring multiple catalogs and tables sitting next to the CRT terminal in order to enter or retrieve data.

THE SELECTION OF CUSTOM PROGRAMMING

The systems analyst and programmer is the professional who can provide the liaison between the businessman and the computer hardware and software. It is he who can take the responsibility to select an optimum computer system, taking into account both the individual needs of the business and the considerations discussed earlier in this series of articles. He is the buffer between conflicting motivations of suppliers and businessmen, and in general he functions as the technical specialist with allegiance to his client in much the same manner as a doctor or attorney. In addition to selecting the system, the consultant can make certain that the system is implemented properly — that all the hardware modules are functioning together and that the systems and applications software are both interfaced correctly to the hardware and are functioning properly. After the system is thus completed, the consultant can proceed with the systems analysis and programming for the custom tasks specified by the businessman.

Certain attributes and actions are desirable in the consultant, relating to the phases of selection of equipment and software, of implementation of the system, of the subsequent analysis and programming of the custom tasks, and of the financing thereof. Each of these phases, in turn, is evaluated in terms of reliability and capability of the analyst.

For the phase of selecting equipment and software, the attributes and actions of the consultant involve his knowledge, his priorities for selection, his clarification of the needs of his client, and his commitment to cost-effectiveness.

The consultant must be capable of extensive knowledge of currently-available hardware and software, complete with a knowledge of mutual compatibilities of all such components. He must consider that compatibility of equipment and systems software with a wide support base usually is far more cost-effective than even more capable components with limited support; that any choice favoring compatibility over capability is nearly always the correct choice; that any capability thus traded off usually becomes available in a short time anyway, if the environment is highly compatible. Expandability also must be included, but usually is consistent with other considerations.

Before he can select the proper equipment, the consultant must have a clear idea of the needs of his client. Extensive discussions of tradeoffs and alternatives, together with probing questions, are necessary for generating such information. Business plans should be probed not only for the immediate implementation, but also for the expected and desired expansions projected two to three years in the future. If the consultant does not take sufficient care in these matters, he may not be as reliable or as capable as desired.

The reliability and capability of the analyst also is indicated by his focus of minimizing the amount of custom programming needed to achieve the tasks desired by the businessman. Commercially-available applications packages must be discussed thoroughly in the context of the business needs, clarifying possible adaptations or modifications, plus required hardware. Such reliance upon applications packages is feasible, of course, only if any needed modifications are not extensive and only if the modified program is well-designed and modularized. The consultant also should select sufficiently powerful hardware and system software to facilitate these custom modifications, and to shorten the time needed for any custom program development in addition to the modifications.

The next phase of implementing the computer system, after the system is selected and purchased, can be done by the supplier, by the consultant, or by the businessman himself. The latter choice is not recommended for any system other than a totally prepackaged system complete with all desired applications software. If the supplier implements the system, it is best not to take delivery and not to pay more than a 10% to 20% deposit until the actual system — not just the same configuration—is demonstrated to function properly in all respects.

If the consultant implements the system, then he is responsible for the correct functioning of the stock system, including all purchased hardware, systems software, and packaged applications programs. Presumably he would have considered the reliability and compatibility of the components he selects, and the competency and reputation of the supplier he utilizes — to ensure support for his implementation, such as warranty repairs. He also would take full responsibility for the correct interfacing of the software to the hardware.

In addition to implementation these stock components, the consultant thus engaged also is responsible to implement the custom modifications needed for the applications packages he selects. Presumably these custom modifications would have had the full assent of the businessman after his being apprised of both the necessity and the costs of the custom work. The advisability of custom modifications would depend upon compelling reasons to justify both the system choice and the choice of applications package, after examination of alternatives such as competing packages or the option of custom programming the whole task.

The financing structure for these two phases depends upon the consultant. The conscientious consultant may well suggest the prudent precaution of a fixed-fee for the selection process, to prevent excessive shopping and learning at the expense of the business. An even better precaution is to have the fixed-fee structure also include the assembly and implementation of the selected components. This prevents costs from climbing unpredictably and uncontrollably from potential difficulties arising from incompatibility or unreliability of the selected components. Since this is the responsibility of the consultant, it forces his experimentation to occur at his own expense. It also motivates the consultant to select the best equipment, especially if he will have to use the same equipment for a month or more for custom programming. This precaution of a fixed-fee, of course, would not apply to the subsequent custom modifications and programming necessary for some applications programs, beyond simple installation.

Part of this arrangement would include the demonstration and thorough checkout of the stock system to verify operation to the businessman, in order to define the period represented by the fixed-fee and complete

non-custom installation. After the system is functioning properly in its stock version, the programmer can begin custom modifications. He is responsible to document his work thoroughly — giving details of the custom modifications and correcting the manuals wherever necessary. This documentation should be provided for both the user and the programmer who will need to decipher his work at a later time.

After the system is functioning properly with the custom modifications, the consultant can proceed with the final phase of systems analysis and custom programming for the tasks specified by the businessman. At this point, the attributes and actions of the consultant become critically important, due to the potential size of the custom programming investment. These involve the evaluation factors of compatibility with the business, reliability, and capability of the consultant.

To ensure compatibility with the needs of the business, the communication between the consultant and the businessman must intensify. Questions of strategy and priorities are paramount, and extensive communication must persist throughout the specification, design, and documentation phase. Subsequently, the coding phase must be consistent with the business needs of economy and speed of implementation.

Questions of strategy arise because most programming tasks, especially the large ones, can be approached in many different ways. One course of action is to create a quick implementation of a custom program which has only limited capability and coherence to the overall set of tasks, but which can create almost immediate benefits. An opposite approach is to delay implementation of the custom program until the whole set of tasks can be defined and incorporated into one comprehensive system with modular design and complete expandability. Such a course would have much larger benefits in the long term at the expense of delay. A compromise strategy would be to envison and specify in general terms the comprehensive system without working out details, then to design and implement the highest-priority portions in a manner consistent with eventual incorporation as modules in the comprehensive system.

The second essential for ensuring that program development corresponds to business needs is persistent and extensive communication with the client. The consultant normally would specify, design, and document the most general, top-level modules of the intended program, then check with the client and redesign if necessary. The process is repeated with progressively deeper and detailed modules, with constant checking with the client about those features which affect operation or capabilities of the program.

Only after this specification, design, and documentation phase has progressed through the deepest modules, complete with checking with the client, should the actual coding of the program begin. Up to this point, the program should be nearly language and machine independent, and the appropriate language may be selected at will. Preferably this selection will be for the highest power language available for the system, provided it is consistent with the coding requirements of the modules, and provided that the need for machine-language subroutines would be minimized and localized. As discussed in the section on compilers, such techniques and choices provide the most cost-effective and expedient implementation of the custom program, ensuring that the custom programming is compatible not only with the needed tasks of the business, but also with its needs of economy.

After the coding for each module is completed, the task of debugging and testing the module begins. Debugging consists of the preliminary detection and subsequent correction of the more obvious errors which prevent apparent proper operation of the program. Frequently ignored, and quite distinct from debugging, is the testing process, which consists of the thorough detection and correction of errors. Testing is initiated only after each module is apparently working without obvious errors, and requires exercising each section of code within the module with all conditions that can affect operation of the code. All extreme and special cases must be anticipated and checked.

The modules in the deepest parts of the structure are the first ones debugged and tested, then the module may be utilized as a part of the next-higher module, which is then similarly debugged and tested. Eventually this process will lead to the top-level program module itself to be debugged and tested, completing the program.

Another requirement for reliability of a program written by the custom analyst is thorough documentation of the program, and this also is frequently insufficiently done. Two types of documentation are needed to accommodate both the programmer and the end-user.

Programmer documentation is essential to enable future maintenance, modifications, expansion, and error correction by other programmers, or even by the author himself months later. Errors may appear even years later, and must be traced in detail — an impossible task without clear explanations of the program design and function. Such programmer documentation should include two formats — a programmer manual and remarks in the code itself. The programmer manual should include a description of overall program design complete with flow charts and module charts, and the comments in the code itself should be generous enough to create understanding both of the coding details and of the correlation of the code to the modules.

The documentation oriented to the end-user should enable the proper use of the program by the businessman. The description of the program and instructions for its use must anticipate the questions of a non-professional and must consider all the special cases relevant to the program. A person totally new to the program should be able to operate the program thoroughly and accurately without difficulty and without any tutoring whatsoever from a professional.

The small proportion of time and money invested in such documentation is an investment in longevity and maintenance of the program which already will have cost substantial sums.

Part of ensuring the reliability of the custom programming, and its compatibility with the needs of the business, is comprehensive evaluation of the capabilities of the consultant in relation to the business. There are several indicators useful to such evaluation. One is his overall approach of the comprehensive design and modularization of the program, as discussed earlier. Another indication is his willingness for persistent communication to establish task specifications and to clarify strategies and priorities of the business. He also would keep these factors in mind throughout the consulting period, and thereby also minimize and control the impact of overruns. Additionally, if the consultant is careful in his design and coding, debugging time, as distinct from testing time, should be minimal. Also, his willingness to test his program thoroughly in the manner discussed above is an indication of his style and conscientiousness. And finally, an indication of good programming practice is orienting the program design and documentation to the end user, ensuring ease of use of the program and clarity and sufficiency of the documentation.

The clarification of priorities is important because estimates of time and cost are crucial to business planning and decisions, yet are likely to be in error. The

CIRCLE INQUIRY NO. 70

analyst frequently errs on the optimistic side, even though he may be aware of this tendency and may try to correct it somewhat. The prudent businessman will plan for the task to cost two to five times the estimate of the analyst, so that the business will not be in the position of having only a partially-completed program which it can neither utilize nor finance further. If the analyst is accurate, so much the better — then more programming may be done.

This type of planning underscores the need for strategy discussions incorporating contingency plans for both early and late completions of the programming. Clear priorities must be set on the desired tasks to ensure that the most important ones are accomplished first, and the program design must be modular so that the most important modules may be implemented and functional, even if some modules remain unavailable for some time.

The total debugging time of the program should occupy only a small fraction of the total time needed for the specification, design, and documentation up to this point. Excessive debugging time indicates insufficient design integrity.

It is important to recognize the clear difference between debugging time and testing time, in the respect that a perfectly functioning program may need no debugging, but still needs complete and thorough testing which may be somewhat time-consuming. There is no shortcut to ascertain that even an apparently perfect program is indeed perfect!

The user orientation of a well-designed and documented program is one of the most visible means of evaluating the consultant, especially for the non-professional. If the program is comprehensive, yet clear, logical, and convenient to use, and if the documentation is readable, thorough, and well-organized, then the programmer probably has done the same in the technical parts of the program. If he is remiss in these respects, or is reluctant to conform to these standards, then beware, for the converse also is probably true.

The job of the systems analyst has been described and evaluated for the selection of components, the implementation thereof, and the analysis and programming of the needed custom tasks. The final question that remains is how to engage these services. More specifically, when should the analyst be hired, and what would be the contract and payment arrangements?

By far the best time to engage the consultant is before any equipment is purchased, so that his expertise may be utilized to assess the hardware and software needs for the desired tasks, avoiding possible costly mistakes of capability, compatibility, reliability, and expandability. Part of his knowledge depends upon discussions with colleagues, and is not obtainable from the suppliers or from the specification sheets — either because of conflicts of interest or because of the technical nature of the information.

It is advisable to hire the consultant in two separate phases, if possible. The first phase is the selection and stock-implementation phase, which is best done on a fixed total fee basis, as mentioned earlier. This prevents overzealous experimentation with possible incompatible or unreliable components which might otherwise promise lower cost or exalted capability — thus it prevents possible cost overruns even before the real work begins. Additionally included in this phase is the minimal analysis and possible top-level programming to generate the information needed for the component selection and for the estimates of custom programming costs.

The second phase is the actual detailed custom systems analysis and programming. This phase would be done on a contract basis either utilizing a fixed-total fee for the complete job, or more likely, specifying an hourly or monthly fixed-rate fee to continue until the job is completed.

The fixed-total fee has the disadvantage of being either too high or too low, unless the analyst is capable of making extremely accurate estimates of required time. If the fee is too high, the problem is obvious, but if too low, the problem is more subtle. One possible effect is programming which is hasty, resulting in error-prone inadequate planning, modularization, and structure. Another effect might be insufficient and inadequate documentation.

The hourly or monthly fixed-rate fee can avoid these difficulties, but is vulnerable to cost overruns. This is mitigated somewhat if the businessman allows for such errors as discussed earlier, and is best controlled by constant communication such that the businessman constantly assesses the progress of the consultant and may take corrective action if necessary — such action presumably would have been outlined in the initial strategy discussions and contingency planning, and would be facilitated by the proper modularization of the program with clear priorities attached to the modules.

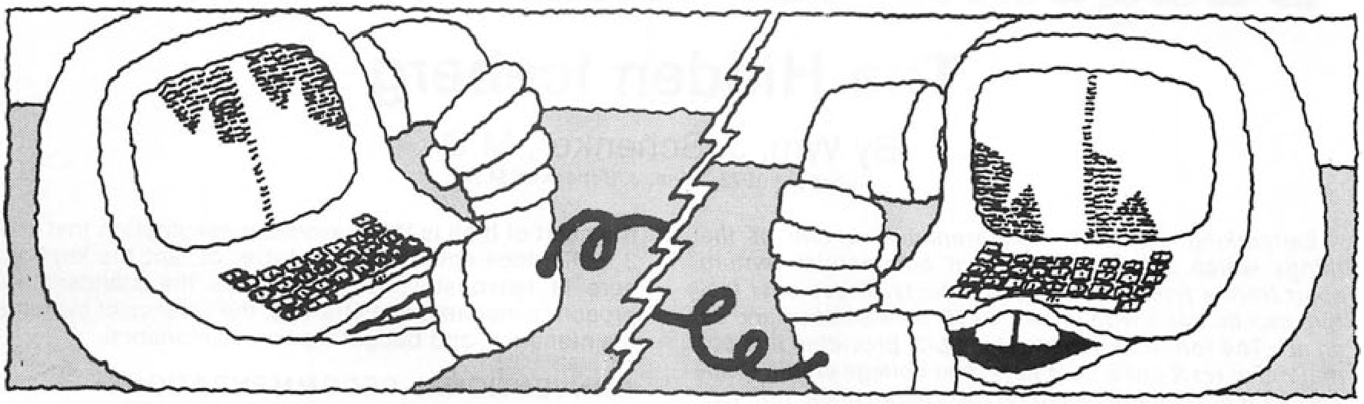
A compromise would be a contract for a fixed number of man-hours at a specified cost. This fixed-time arrangement eliminates the possible excessive cost or haste of the fixed-total arrangement, and limits the overrun vulnerability of the fixed-rate arrangements. The consultant knows beforehand what his monthly pay will be and what his time limits are. He can then utilize priority and contingency planning to achieve the optimum capability by the end of his consulting period.

It is of course expected that throughout such consulting periods the consultant would keep a detailed log of his hours and activities. Such a log not only would document the time utilized, but also would categorize time spent for conferencing, specifying, designing, documenting, coding, debugging, and testing the custom program. These breakdowns would be useful both to the consultant in evaluating his own performance and to the businessman in appreciating the broad spectrum of work necessary in programming, beyond mere coding of the program.

Finally, the issue of payment arrangements arises. Of course, as with any other purchase of goods or services, either the computer, the consultant, or both may be paid in cash or may be financed, leased, or lease-purchased. An additional option, tailor-made for such a situation, is the bundling of the hardware, prepackaged software, and custom systems-analysis into one lease or lease-purchase package through a third-party leasing company. This is especially available if the consultant is utilized from the beginning for the selection and implementation of the hardware and packaged software. Frequently the consultant could specify the cost of the components, then add his fee to include both a fixed-total fee for selection and implementation, and a subsequent fixed-time fee for a number of man-hours for custom analysis and programming. This combination would create a fixed total price which would then be financed by the leasing company for a three to seven year period. A total package including all hardware, software, selection and implementation thereof, and two man-months of programming, which might cost anywhere from $8,000 to $20,000, could be financed through a leasing company for anywhere from $250 to $500 per month.

Such a lease-purchase arrangement has the advantage of tax writeoffs throughout the lease, and possibly even investment tax credits passed through the leasing company on the subsequent purchase — provided that special conditions are met as specified by the IRS. □

TELECOMMUNICATIONS software for the AM-100



Do you have an AM-100 or sell them? Wouldn't it be nice if it had remote timesharing capability? WELL NOW IT DOES! Wouldn't it be useful if it could interact directly with other computer systems? WELL NOW IT CAN!

Khalsa Computer Systems has completed a powerful, easy to use telecommunications software package for the D.C. Hayes modem card running with the Alpha Microsystems AM-100. Add a phone line and a data coupler (DAA), and you have the capability to interact with virtually ANY COMPUTER or TERMINAL IN THE WORLD at up to 300 baud.

IT ANSWERS THE PHONE . . . A remote terminal with a standard modem may call your computer and use it just as a local terminal would. Passwords on user accounts protect your system from unauthorized use. Applications:

1. You can actually provide a REMOTE TIMESHARING SERVICE for subscribers and make money doing it.
2. You can have an AM-100 at your office and use it from your home.
3. You can keep your AM-100 at home and still use it from work!
4. You can enable or disable the dial-in capability of a single line or of all the lines anytime, from your terminal, so your system can conveniently be made 'accessible' or completely 'secure'.

IT MAKES PHONE CALLS FOR YOU . . . You may have your AM-100 call a remote terminal and have it connected as if it had called in. You may have your computer system CALL ANOTHER COMPUTER SYSTEM, allowing you to use the entire AM-100 as an intelligent terminal on the remote system. Think of it:

1. You can use the AM-100 to develop large programs 'off-line', then submit them to a remote system by phone when you are ready. You probably know that massive on-line editing on a large timesharing system can be costly. (You can send single or multiple files from your AM-100 system to a remote system.)
2. You can have the AM-100 collect data via card, paper tape, or optical character (OCR) readers, keyboard terminals, or lab instruments, PRE-PROCESS the data to eliminate errors, reformat the data, etc., then send it on to a larger system for further processing. The telephone company has done this for years with the phone repair data it collects in the field on mark-sense cards.
3. You can acquire data files from a remote system, storing them on your AM-100 disk.
4. You can even record every character of a session on a remote system for later editing or study . . .

IT'S DONE RIGHT . . . Our telecommunications package is VERY GENERAL in scope. It requires no special software in the remote systems you connect to. It is completely compatible with the Alpha Micro operating system, as we provide an interface driver (IDV) for the modem board and use the operating system for all data transfers. This preserves COMPLETE DEVICE INDEPENDENCE. You may have as many incoming or outgoing telephone lines as you need, and attach to them whatever terminals you like! We provide corrections and enhancements in the form of updates. We plan to put the AM-100 on the PCNET (a worldwide personal computer network) in the near future. We think you'll be pleased with our documentation too, as we give explicit instructions on installation, testing, AND usage.

FREEBIES . . . To make it even easier for you to say "I'll take it!", we are also providing several valuable extra programs for FREE. One allows users to send single or multiple line messages from any AM-100 terminal to any other – with the source terminal being automatically identified at the destination terminal (a la PDP-10). Another lets you get a directory of all the files on a disk, but writes the output to a FILE instead of just to a terminal. Another is a DISK CRASH RECOVERY program, which helps find and restore files, even if the master or user file directories are destroyed. Finally, we're throwing in vital details about QDT and DSKDDT, two very powerful debugging programs which Alpha Microsystems hasn't gotten around to documenting yet . . .

WHAT ABOUT AVAILABILITY AND COST? Both the software and the necessary hardware are IN STOCK NOW. The complete source code for ALL the programs is available for (a very reasonable) $300. Object code alone is only $125. Add $275 if you want the D.C. Hayes modem card too ($279.95 when purchased separately). We are proud of this package, and we are sure you will be pleased with it too. Call or write to receive full details, or better yet, **COME SEE IT WORK!**

Come See Us

KHALSA

COMPUTER SYSTEMS INCORPORATED

500 South Lake Avenue, Pasadena (213)684-3311
Tuesday-Friday 12 noon to 8 pm, Saturday 12 noon to 6 pm
Closed Sunday and Monday

Khalsa Computer Systems, Inc. has also been known as the Byte Shop of Pasadena since we opened in 1976.

Budgeting for Maintenance — The Hidden Iceberg

By Wm. J. Schenker, M.D.

Earmarking capital for maintenance is one of the things which sets a business or commercial venture apart from a typical hobbyist's activity. These cost factors can be dealt with in two ways, as a science and as an art. The former is the most visible, providing subject matter for textbooks, seminars, and college credit. Complex as it is, it is still based on the simple linear premise that 2 + 2 really does equal 4, and such like. This makes it relatively easy to read and write about.

On the other hand, what the science deftly avoids is a large object whose icy tip spells disaster in the deep for the unsuspecting businessman or professional. Etched in large letters on the submerged surface, hidden to ordinary view, is the warning, "Murphy's Law (and Cohen's Corollary) Reigns Ever Supreme!"

This murky subject, the center of many a hallway conversation among insiders, and rarely discussed as the computer systems vendor plies his trade among the innocent, will be emphasized in this article, along with several maverick points of view. Together they make up what may be called the art of maintenance budgeting.

INTRODUCTION

To the hobbyist, system failure is a challenge which promises at worst a broadening knowledge, at its best the ego rush of successful debug and repair of a non-working conglomeration of wires, chips and boards. This challenge is not constrained by time factors nor by responsibility to an outsider such as a customer. Since time is money you can see how hobbyists can spend $10,000 in labor to repair a $1,000 computer.

To the businessman or professional, however, equipment breakdown means an added expense or lost income and a blemish on the organization's market image. To appreciate the significance of this, consider the following. When a computer system suddenly stops running ("goes down" or "crashes" in the vernacular), the obvious cost is loss of service to the customer. What is probably in the long run a much more telling loss is the loss of data base integrity. This occurs when the transaction in process at the time of the crash gets lost (or duplicated!), a record during update is lost, or a spurious record pointer change occurs. This last can result in possible loss of a massive number of records. The bottom line effect of all this on a business is loss of customer confidence.

Look then at what must be done to maintain uninterrupted performance for a customer or client at the level the latter is accustomed or contracted for. This way one gets some idea of what these actions will cost in the way of capital investment and added payroll.

The problem can be seen to warrant close study and inspired application, indeed so much so that journal articles and textbook chapters are dedicated to it. This literature tends to follow the formula of computer science publications in general. The orientation is extremely rational in tone and the reader is assumed of a likewise bent. The people in this field tend to have an orientation or background in mathematics. Thus the literature is also based heavily on a mathematical or statistical approach worthy of the physical sciences. At the heart of it all is the reasonable assumption that 2 + 2 really does equal 4. As a matter of fact the keyword here is reasonableness. It typifies the standard approach to maintenance strategy, the science of systems maintenance, and budgeting for maintenance.

CONVENTIONAL RECOMMENDATIONS — AND UNCONVENTIONAL COMMENTS

A superficial appraisal of maintenance costs would focus on the obvious: the specs of the warranty and maintenance contract, those of the latter usually including location of service depot, minor or all parts covered, charge for travel time, preventative maintenance schedule, and service agreement on an hourly ("on call") or contract basis.

However, experience dictates considering also the design and configuration of the system itself because these decisions, made long before system purchase, have a heavy impact on subsequent problems. Accordingly, note the following brief but representative list of recommendations an end-user is typically advised to investigate prior to purchase. Then following each of these points note my own often rather unconventional recommendations under the headings, COMMENTS.

BUY FROM ONE MANUFACTURER

A mixed-brand potpourri will find at the time of breakdown the inter-vendor finger pointing, each one accusing the other as the basic culprit.

COMMENT. In the first place, when you're dealing in small systems, never buy from ANY manufacturer. Buy instead from your local retail computer store. People you can talk to on a first name basis and whom you're likely to bump into at your neighborhood supermarket are much more likely to be responsive to your needs. In the second place, if you buy what are called S-100 products, you can count on a relatively high degree of inter-brand compatibility. More on that later.

MANUFACTURER PEDIGREE

The brand should be big-name and well established, even though the initial price is much higher. Avoid the fly-by-night and those without a track record.

COMMENT. There are no big name manufacturers whose primary business is small systems. Avoid the minicomputer firms and the big semiconductor manufacturers for whom micro systems are only a sideline: they're too big to care about you.

Besides, the "biggies" don't necessarily use the latest technology (because they feel somewhat immune to market pressures?). For example, the best support chip technology today is Low Power Schottky (LS, for short). It is more reliable than its predecessor TTL because it produces less heat and electrical noise.

VENDOR "BURN-IN"

Burn-in is the process of pushing the hardware close to its limits to see if it will stand up to prolonged temperature, vibration, humidity, dust, and electrical noise stresses. Investigate the details of this procedure by the manufacturer of your choice.

COMMENT. Temper the standard advise with the fact

that the modern chip technology described just previously tends to be quite reliable once you're past the "infant mortality" stage. (This is chip failure within the first several hours of use.) If you do feel you need it for your application environment, don't rely on the micro manufacturers. They do very little of it. Instead have the local retail store where you buy your system do it for you.

MTBF & MTTR STATISTICS

MTBF is the mean time between failures, and MTTR is the mean time to repair such failures. Ask the manufacturer to show you his figures.

COMMENT. Ah, the beauty of those neat MTBF and MTTR statistics! So crisp, so scientific, no neat, so precise. So valuable?

At the outset the one thing to keep uppermost in your mind is that these MEAN figures apply only to Mr. MEAN End-user. He sits right in the middle of the bell curve of probability. Way off to one side of him is the fellow whose equipment never breaks down. But off to the other side is the fellow whose equipment fails before he even gets it out of the packing carton. No manufacturer will ever guarantee that you'll always fall between Mr. Mean and Mr. Superlucky, but many of their salesmen will. Not in writing, however. (See Figure 1.)

To put it another way, if you need to rely on 5 years between breakdowns, you can't pick a system with a MTBF of 5 years (even if such were available). You'd need a system with a MTBF of FIFTY years. To ignore this unpleasantry is courting a case of the "pre-demo blues" and a full-blown operation of Murphy's law.

MANUFACTURER WARRANTY

Investigate carefully the warranty details. Consider this an important step.

COMMENT. In the small systems field the best thing you can do with your factory warranty is to place it under the Presto log in the fireplace. It will start faster. They're worthless in terms of turnaround time (up to three months) — another reason to buy local. But apart from this consider the following. If you're involved in marketing a service, you're fully aware of how much the "personal touch" or its absence can mean to business success. How much personal touch do you think you'll get from your system's manufacturer if your system's serial number is 14857?

SOURCE OF MAINTENANCE CONTRACT

Buy your maintenance contract preferably from the system manufacturer, or as a second choice, from one of the nationally-known service companies.

COMMENT. Hardly any micro manufacturer offers one, and the national service companies don't have the necessary experience with small systems. Again buy local, including your maintenance contract.

An interesting point about service contracts in general is that the usual service availability is Monday to Friday, 8-5. (With travel time that often ends up 10-3). But what if you're starting an important project well within the capabilities of your micro system? Under operation of Murphy's and other perverse laws, it would not be that outlandish for your system to crash Friday evening at 6 P.M. — with service unavailable until Monday morning.

TRACK RECORD

Buy a system that's been out in the field with a long, reliable track record.

COMMENT. This particular recommendation is a tricky one. It's right, up to a point. However, the entire small systems market for professional applications is barely two years old. Is there then any other factor the

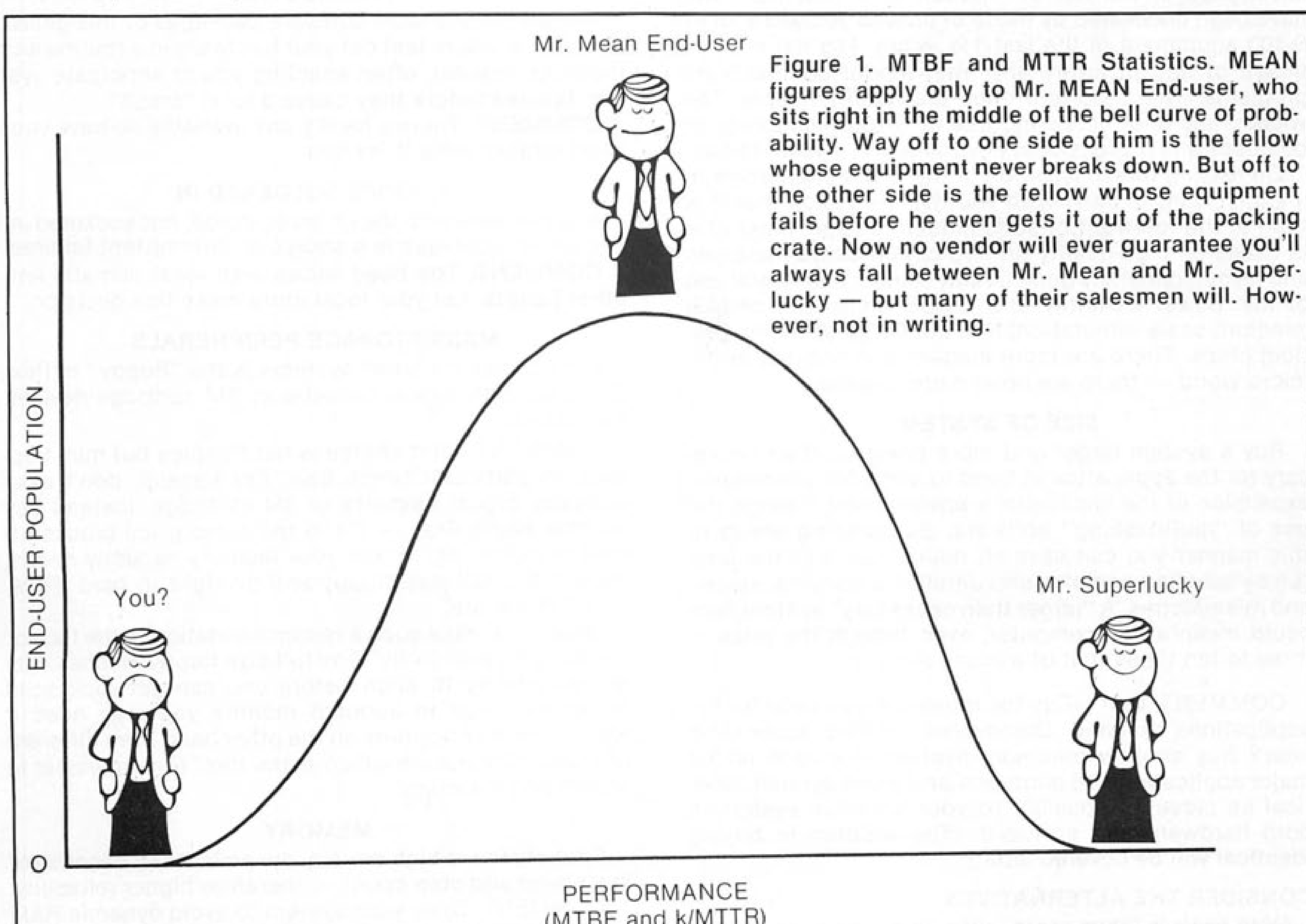


Figure 1. MTBF and MTTR Statistics. MEAN figures apply only to Mr. MEAN End-user, who sits right in the middle of the bell curve of probability. Way off to one side of him is the fellow whose equipment never breaks down. But off to the other side is the fellow whose equipment fails before he even gets it out of the packing crate. Now no vendor will ever guarantee you'll always fall between Mr. Mean and Mr. Superlucky — but many of their salesmen will. However, not in writing.

cautious buyer can substitute for the missing ingredient? Product maturity? Very definitely.

PRODUCT POPULARITY

Buy a "popular" system, or as a micro computer tech I know said recently, "When you're in small systems, it's good to have 'friends' around." In other words, avoid a system which has only 500 units out in the field. You want a system using boards and subsystems or modules which are second sourced (i.e., compatible enough with another vendor's product to be directly replaceable by it). But even more important, the equipment should have off the shelf availability at your corner retail store. Keep in mind the above recommendation on the basis of present market conditions and products from one or several of the S-100 manufacturers.

This is probably as good a point as any to digress a moment and fill in a brief sketch of the S-100 phenomenon. Frowned upon with disdain by the biggies of the mini world (and even by fellow microcomputerists sold on their own unique systems) when it first arrived, it is now ubiquitous enough that it's being seriously considered for acceptance as an ANSI and IEEE standard. It was originated by MITS of Altair™ fame and marketed in early 1975 at the then unheard of price of $400 for a "complete" computer in kit form. So popular did it become that a bevy of other small manufacturers soon climbed aboard and began marketing Altair-compatible products. The Altair bus then soon became known as the "hobbyist bus" and finally, because of its 100-pin configuration, the "S-100 bus".

The S-100 products on the market today certainly are not panaceas, but in the hands of a competent retail store a good selection can be assembled into a system as reliable as one from the large manufacturers. To be sure, this involves a special expertise: an intimate knowledge of the list of "hidden little gotchas" that have been uncovered by those of us who've used a lot of S-100 equipment in the last 1½ years. The list is composed of design errors and inter-vendor compatibility problems. The important ones are readily fixable. The widespread and increasing use of S-100 technology in business and professional environments attests to this.

On leaving this discussion of the S-100 phenomenon, it's important to carry with you the realization that it is right at the forefront of solid modern design: most of it is based on high-quality printed circuit board materials and layout, generally good circuit design, and liberal use of low power Schottky and CMOS chips and of MSI (medium scale integration) and LSI (large scale integration) chips. There are more elegant bus designs in the micro world — there are none more popular.

SIZE OF SYSTEM

Buy a system larger and more powerful than necessary for the application at hand to allow for subsequent expansion of the applications environment through the use of "multitasking" software. By planning ahead in this manner you can save on maintenance in the long run by avoiding complex and unreliable retrofits, kluges and mismatches. A "larger than necessary" system here could mean a minicomputer, even though the price is three to ten times that of a micro system.

COMMENT. Don't. Buy the minimum you need for the applications at hand. Discovered another application later? Buy another minimum system. For each added major application add another stand alone system, identical as close as possible to your previous system in both hardware and software. (The wisdom in buying identical will be covered later.)

CONSIDER THE ALTERNATIVES

With today's labor costs in the form of programmer's wages, custom software implementation of the multitasking you'll need can cost five to ten times the equivalent in hardware.

MORE ON MULTITASKING

Just to ensure proper understanding of this point a moment's digression into some details is necessary. This technique allowing two or more applications to be handled by one machine, at times apparently simultaneously, actually is a process akin to juggling 87 balls in the air at once; the computer's software is interweaving the multiple applications or "jobs" one between the other. The technique harks back to the days when the only hardware available were maxis and minis which cost so much per system that you were easily persuaded to squeeze out the maximum performance per piece of hardware bought.

In full blown versions used on the biggies, it involves developing techniques for "queueing" one job after another according to priorities, error checking of a complex nature, and complicated "rollback and recovery" of data when the system eventually crashes. It is responsible for a large software "overhead", i.e. software which is not earning you any money while using up your computer's resources, both memory and processing speed or "throughput". Furthermore this is the kind of software that can take man years to develop and therefore costs plenty. It also helps to make the science of software development mysterious and their practitioners irreplaceable. Further, one would have expected by now something that complex and expensive would have made the large central computer systems "bullet proof" or impervious to error. But it is certainly in part responsible for the somewhat unsavory reputation that computers have earned in non EDP circles over the past 20 years.

"DIAGNOSTICS"

Buy all the available software packages of this genre. They allow you to test out your hardware in a routine and thorough manner, often enabling you to anticipate system failures before they cause a total "crash".

COMMENT. There's hardly any available so have your local retailer write it for you.

IC CHIPS SOLDERED IN

Buy systems with the chips soldered, not socketed in. The latter technique is a source of intermittent failures.

COMMENT. The need varies with local climatic and other factors. Let your local store make this decision.

MASS STORAGE PERIPHERALS

First choice for small systems is the "floppy" or flexible disk, with digital cassette or 3M cartridge devices for backup.

COMMENT. First choice is not floppies but mini floppies, in particular North Star. For backup, don't even consider digital cassette or 3M cartridge. Instead buy another North Star — it's in the same price bracket. If mini floppies don't meet your memory capacity needs, bypass the full size floppy and go right to hard disks. You'll thank me.

How can I make such a recommendation in the face of the floppy's popularity? The full-size floppy mechanicals are very tricky to align before you can get rock solid reliability. Then in about 6 months you may need it again. The mini floppies on the other hand have different physical dimensions which make their tuneup easier to obtain and maintain.

MEMORY

First choice is high density dynamic RAM, because of low power and chip count — therefore higher reliability.

COMMENT. Spec your system to avoid dynamic RAM — it's too flaky with high speed peripherals using DMA

(direct memory access), such as floppies and some graphics displays. Order instead fully static boards. Also avoid super high density boards such as 64K or 32K boards. Spec your system in increments of 16K. That way, if a chip goes bad on one board in a 64K system, you can still operate in a degraded mode with the other three.

PRINTER SPEED

This is not usually considered in estimating maintenance costs. In general buy a fast printer that you can reasonably afford. (Low speed in the big-name world means about 200 characters per second or somewhat less than 200 lines per minute. Medium speed is about 600, and high speed goes up to an astronomical 30,000!)

COMMENT. How much speed does a printer really need to have in a small business environment? How slow is too slow? Let's eavesdrop on a small businessman or professional who's saddled himself with a printer "too slow" for his needs. The suprising and significant discovery is that he's chafing at the bit, waiting for his printer to finish its output, typically not because he urgently needs the hard copy. If in fact he's in that much of a hurry for his paperwork, one of the following management faux pas may have occurred.

1. His customers are in the habit of dropping in unannounced to pick up their 30-page reports. In this case some customer appointment scheduling may be in order.
2. If, on the other hand, the press for time develops because his customers have contracted for 500 to 1,000 page reports, there may be something basically and radically wrong with the way the job was spec'd out in the first place. There are presently hundreds of Fortune-500 companies responsible for chopping down stand after stand of pulpwood forests to add their contribution to word pollution in the form of unreadable (and usually unread) reams of computer output. This is often the yardstick by which many naive managers measure the worth of their EDP departments. This is one custom of the biggies that the small businessman or professional would do well to avoid. Learn (and try to persuade your customers) to format your text output with the fat surgically excised and most of your statistical data in graphics form.

However, when all's said and done, you may have on your hands an application which requires the entire documentable data stream to be handed over to the customer in hard copy form for legal or other reasons. Here you're going to have to settle for a printer at the middle or high end unless the reports are few and far between.

You'll find the typical computer entrepreneur hovering over his printer not because he needs the output but because he needs the computer. He needs it back so he can get on with more processing, like accepting more input. The standard answer to that dilemma is: buy a faster printer. Look, however, to the Further Excursions into Unorthodoxy section describing further heresies for an alternative that's cheaper and that increases overall system reliability. Remember, your motto in connection with printer speed versus maintenance problems should be, "Speed kills!".

INSTALL A UPS

This means an uninterruptable power supply. It should be large enough to handle the power requirements of your entire computer system.

COMMENT. Yes.

AMBIENT TEMPERATURE

Keep your computer system cool; use adequate convection in the form of fans inside the chassis, and air conditioning in the room.

COMMENT. Yes.

NICOTINE

Avoid smoking in the computer room. The fumes are poison to magnetic disk and tape media.

COMMENT. Yes, it prevents computer cancer.

AC POWER LINES AND GROUNDING

Use good filtering and solid grounding procedures, respectively.

COMMENT. Yes.

MODULAR VS ALL-IN-ONE PACKAGING

It's OK to buy the latter. It cuts down on troublemaking interrconnect cables, dust,meddling by unauthorized personnel, and generally makes for a neater appearance.

COMMENT. Instant death. This is what all-in-one spells from a maintenance point of view.

There is a tendency abroad of late, especially in the small computer field we are dealing with, to package entire systems in one neat tidy little enclosure which includes keyboard, CRT, computer, mass storage devices, and even in one case at least, the printer. The enclosure typically has futuristically shaped round corners, and aerodynamically designed contours. In short their appearance would harmonize delightfully with all the props on the "Star Wars" set. What this invites in maintanence terms, however, is sudden death. We can hark back to an old fashioned "putting all your eggs in one basket".

Pardon me for waxing poetic here — I consider this to be one of the more pernicious trends in the contemporary scene. True it cuts down on inter-module connections — but at what cost when the inevitable breakdown occurs? Although the janitorial service will love you for such a purchase (it's just a dream to dust off that one big box with all those round corners), and although you can keep various non-authorized personnel from messing with the interconnects and the guts — consider the price when the moment(s) of truth comes.

Consider the cost to you who pays the bill, until that time arrives when entire systems are:

(1) as cheap as throwaway calculators and
(2) as reliable as light bulbs.

Consider the cost when you understand that the most common cause of breakdown (in a system properly designed and installed in the first place) is mechanical failure. Failure of front panel switches, keyboard switches, disk drive mechanicals and assemblies, tape drive mechanicals and assemblies, printer mechanicals, edge connectors on printed circuit boards, copper traces on these same boards, and switches and controls on CRT terminals to name the major categories except one, interrconnect cables and connectors. This last category of failure is markedly reduced by an all-in-one design. This is nowhere near enough to counterbalance the other categories.

FRONT PANELS

Your system should include this item. It's handy in troubleshooting (and incidentally helps in software debugging).

COMMENT. Avoid them like the plague. The only one who might want one is your maintenance man, and he'll leave it in his toolbox on many jobs.

TLC DURING INFANCY

There is another factor, not apparent to the newcomer in the field of EDP, which should be considered for its impact on maintenance costs. Although a system should operate in flawless fashion the first time it's powered up, that almost never happens. A computer system, like people, needs lots of tender loving care during infancy to ensure getting started on the right foot. Lacking this care, troubles will hound the system pos-

sibly to its grave. So until this stage is reached, you can't consider visits by service personnel as part of maintenance costs and loss. When the system is completely debugged and performing to vendor specs for a month, subsequent breakdowns are then properly in the category of "downtime", or failure.

COMMENT. Another reason to have service personnel close by, which means your local computer store again.

FURTHER EXCURSIONS INTO UNORTHODOXY

"ON LINE" AND "REAL TIME"

Computer people bandy about two terms you should become familiar with, on line and real time. There are many definitions extant, but all you need remember is that some applications allow the computer to work in spurts with long pauses for resuscitation in between, and some applications require the computer working 24 hours a day, 7 days a week. The closer your application is to this continuous type of affair the more on line or real time it'll be considered.

A piece of advice. If you're planning an application which could classify as pretty much on line or real time, stop and reconsider.

"NON STOP"

Suppose your application calls for nothing less than the extreme, a continuous run? It's then, logically enough, labeled non stop.

More advice. If you're planning this kind of an application, stop and don't reconsider.

Unless your retail store can configure your hardware and write enough software to give you the micro systems equivalent of what Tandem Computers, Inc. claims in their ads they can do, don't try.

THE NITTY GRITTY

Which brings us into the nitty gritty of my unorthodox approach to budgeting for small systems maintenance. My prescription for the typical business application which, although not non-stop does have significant deadlines to meet, is simple.

Buy two of everything. Two complete systems. Two computers, two sets of identical software, two sets of dual mini floppies, two backup storage devices (mini floppies again), two keyboards, two video monitors, two printers, and lastly two complete sets of interconnect cables. (Ignore this item and the whole deal is off.)

This strategy has three things going for it. The first is the ability to keep operating during a critical phase of activity when one computer crashes. True, you lose the data that was in transaction, and you can lose records. But with proper mass storage backup, that barb can be dulled. Just flip the switch of the other system, and transfer your work to it. (If you get your retail store to write some software and add some minimal hardware you can get a semi-automated transition from one system to the other.) You've spent twice as much in capital outlay in exchange for almost instant repair service, unobtainable any other way at any price.

The second advantage has to do with that low speed printer recommended earlier. By using your backup computer that is otherwise idle to drive your printer, you can overlap your application functions in time. For example, you can be talking to one computer via its keyboard while the other is printing out your three hour report. In this manner you can usually do quite well with a 15 CPS (characters per second) printer where a 60 CPS would be considered the bare minimum, or a 30 instead of a 120, etc. (This same strategy can be considered if you're trying to get by with a low speed mass storage peripheral, the audio cassette, which can be quite reliable if properly set up.)

The third and most important advantage I save for last. If is the key element in a novel concept of computer technology, geared to match the microcomputer's role in the increasingly popular EDP *trend towards "distributed intelligence"* through "distributed processing". Instead of relying on one large computer to do all your work, you spread out some of this work by using a network of small computers scattered around the hinterlands.

In effect you trust your eggs to more than one basket. This trend is having a salutory effect on the whole industry, making systems less vulnerable to "total crash". When the big one goes down, people out in the boondocks can still do some work while waiting for the system to come back up again. It's cheaper. It's also less complex and mysterious.

DISTRIBUTED MAINTENANCE

I recommend we start doing the same thing now with the maintenance process. We need what I like to call distributed maintenance. Spelled out this means that instead of relying on a central repair source (the manufacturer, or a national repair organization) we get this service out into the field as close as possible to the one who signs the bottom line, you the end user.

DISTRIBUTED MAINTENANCE PROTOCOL

SETTING THE STAGE

The success of this strategy will depend on your local computer store for three things.

1. A contract for repair of the defective equipment you track down with this method. It should specify the maximum "turnaround time" you think you're application can tolerate. One to three weeks will usually do for typical applications.

2. Developing software in the form of a short diagnostic package that will tell you when in the course of your testing you've bumped up against the troublemaker and have the system runnable again. This message will ordinarily be in the form of video screen prompts.

3. Installation of an electronically simple yet very effective monitor that tells you if your power supply secondary voltage outputs (usually 3) are in good health. This will be in the form of little red pilot lights set into your (otherwise blank!) front panel. Or they can be installed inside the computer cabinet away from the high voltage end of your power supply, which should have a protective barrier placed over it by the store personnel. This is so you can't monkey with it accidentally on purpose. Using this last arrangement the lights should be in clear view on lifting off the computer cover.

The importance of these monitor lights is this. If any of them are out, it means you've lost one of the voltages. You should proceed no further with subsequent tests but get your maintenance man on the phone. Fortunately this will be a rare occurrence.

First phase. Swap modules or subsystems, one by one, from the known good to the crashed system. Be sure the power is off both systems as you're making each swap. Off how long? Long enough for the capacitors to discharge when the blower fan stops rotating. Do this until you find the trouble. Here's a good sequence to follow: interconnect cables, mass storage device, video monitor, keyboard, printer, and finally the computer itself. By now you'll have found out which module is the problem. If it's any but the computer, it's a job for your retail store. If it is the computer, go on to the second phase.

Second Phase. Remove the cover from each of the computers. Start swapping boards one by one in the same manner you did with the modules previously (remember, power off!). A good sequence to follow: memory boards, mass storage interface board, I/O board, and finally your CPU card or board.

Simple Procedure. Remember, don't make this a complicated procedure. Do it "by the numbers", preferably written down on a large cardboard placard placed on the wall near your system. Teach yourself the technique first, then your secretary, nurse, bookkeeper, office boy, or Girl Friday. A maximum of maybe ½ hour using almost no technical expertise, is all you'll need to track down most troubles that can arise. You can then, in a more leisurely fashion, send the defective piece of equipment to your retail store where routine (and thus less costly) repair at the component and IC chip level can be performed.

Equivalent Performance. To achieve the equivalent performance in terms of ultra short downtime and repair of defects would require keeping a full time computer tech on your premises (and payroll), and he would have to have available a full set of replacement parts to achieve his goal, almost the equivalent of "two of everything". The salary of that tech in today's market is $15,000 a year and up.

Cheaper in the Long Run. Distributed maintenance, made possible by modern technology, can move computer science considerably closer to the kind of performance you, the businessman or professional, expected to get in the first place. It's based on a computer system that's almost always working.

Isn't there anyone who can afford to get by on a one-computer system? Yes, there is. He is the man wealthy enough to afford the luxury of losing customer confidence in his operation. For the rest of us, the equation more properly reads: "One computer = half a system." For us it's cheaper in the long run to be able to build and keep customer or client trust in our business or professional operation, for this is what translates to more income.

If this point does not gain widespread acceptance among the growing ranks of small businessmen/professionals considering the use of computers commercially, an interesting repeat performance in the history of data processing is likely to occur. During the '60s and '70s many medium sized and large businesses got badly burned in the expectation that computer systems would, easily and universally, live up to the claims of almost continuous operation. During this period the small businessman's blessing in disguise was simply that he couldn't afford to take the plunge in the first place. What we may easily encounter in the next few years is a whole new generation of disillusioned entrepreneurs, this time based on microcomputer unreliability rather than big system problems.

No Fancy Test Equipment. While on the subject of cheaper, let's cover a related point. Don't be talked into buying fancy and expensive products that enable you to troubleshoot like the pros do it. That's not what distributed maintenance is all about. It's about non electronic people using their time to get on with their own profession. Don't buy a scope, a logic probe, or a logic analyzer. And don't get into swapping IC chips either. You could blow a good one after a bad one that way.

MAINTENANCE IN A SMALL TOWN

With all this reference to relying on your local retail computer store, what if you're in a town with no such source available? In that case what might otherwise appear to you as a luxury, distributed maintenance, becomes a stark necessity.

One Proviso. Even with two of everything don't even consider a near-non stop application if you're located in places like Last Chance, Kansas or Winnemucca, Nevada. There you'll need three of everything. Also, you'd better consider making a special (and costly) contract with the nearest retail store with provisions for (a) the technical personnel remaining on the site until the system's through its infancy and TLC period, and (b) paying them only ½ the total purchase price on delivery, the other half after certain clear-cut, mutually agreed upon tests can be passed by the system's operation. If you can't get this kind of arrangement consider (a) foregoing the pleasures of rural life and moving to or near an ugly big city or (b) running your business as before, in the manual mode, and buy a cheap computer for use at home, calling it a hobby.

THE WEIRDEST RECOMMENDATION OF ALL

This one's saved for last, since it's so obviously beyond the pale in our culture where rationality is considered the final criterion of any scientific endeavor.

Right at the outset of systems planning, before you even look at your application needs, step back and ask yourself this question, and then answer it as honestly as you can. "Am I (a) consistently lucky in business or technical ventures, (b) lucky as often as unlucky, or (c) consistently unlucky?"

If the answer is a, then much of my ramblings can be ignored. If it's b, it'll pay you to reflect on them. If it's c, you're in the same boat I'm in, and to ignore my warnings augers well to bring you deep grief and near insanity in the form of slipped schedules and broken promises.

SUMMARY

This article has emphasized the more quirksome aspects of small systems maintenance problems. The major points made were:

1. Cost-effective maintenance decisions depend heavily for their success on making the correct choices long before the system goes down. As a matter of fact, it is at the time of original spec'ing out the system to be purchased that most of the die is cast.
2. The single most important aspect of the system specs includes:
   a. depending on purchase of 2 of every piece of equipment, thus allowing
   b. the full exercise of the distributed maintenance concept.
   Spelled out for small systems it means doing in-house swapping of interconnects, modules, and boards, backed up by a firm contract with a local reputable computer retail store for actual repair of the faulty equipment.
3. Some contentions, mostly unprovable, were discussed regarding the phenomenon of luck or chance in relation to technology.

HOW TO EVALUATE THIS ARTICLE

After reading this far, if you're a businessman or professional concerned about how he spends his dollars, you should be asking the question, "Sure this presentation is witty and provocative, but does he know what he's talking about?" If you ask this question, you've got to ask the next question, "Who can I ask to get the answer to my first question?"

The Experts. You're left to turn to the "experts" and the "authorities" in the field. Several things to remember about these fellows, however.

1. The field is so new there hasn't really yet developed a large cadre of knowledgeable people.
2. There are those experts whose expertise is in the mini and maxi computer field. Therefore they're likely to give you the conventional list of recommendations that are valid in their world. Ipso facto, most of my unconventional recommendations will be "thumbs down" for them.
3. Of those real experts we're looking for there are two general categories:
   a. The ones who've decided to capitalize on their know-how. They become manufacturers and computer store proprietors.

b. The other kind who are hard to find. They're usually not in the market place selling their know-how but busy in their labs having fun. You won't find them in the yellow pages.

The 87-foot Monster. But wait, there's more yet. Filling this void then is the 87-foot monster I've been making the butt of my argument throughout this article, the manufacturer's salesman. He's dangerous for your computer system's welfare because:

1. He's more accessible. You'll run into him everywhere: at the trade shows, on TV and radio commercials, and in the slick business magazines.
2. He's got less scruples.
3. Mathematical and statistical "facts" are great tools in his hands to blur your vision of the nitty gritty you should be appraising instead.

The Saving Grace. With all the caveats just mentioned, where then can you turn? The answer is actually pretty straightforward. Go to your local retail store and ask for a list of the purchasers of 10 or 12 complete business or professional systems they've installed prior to 6 months ago and are presently serving on a full maintenance basis. If they haven't got that many to show you, chances are you shouldn't be doing business with them. They haven't got the experience. Contact those businesses on the list and inquire about maintenance experience and costs with their system.

COHEN'S COROLLARY

I've enjoyed writing this article and hope you have enjoyed reading it. But the alert among you may have noted there's still one item mentioned in my introduction that has yet to be laid to rest, namely the enunciation of Cohen's Corollary to Murphy's Law. You can't have heard of it before because this is its first public proclamation. It goes like this. "When you've taken that very last precaution possible to prevent Murphy's Law from operating in your applications environment — THAT'S when it probably will."□

ABOUT THE AUTHOR

The author has been a practicing physician for 22 years and became interested in computers in 1974. After completing a course in computer technology at Diablo Valley College, he became active in medical applications of microcomputer systems, in particular, S-100 systems. The latter includes membership on a team which developed a network of micros to implement a large clinical laboratory system, and most recently, installation of a word processing system used by doctors to generate the patient visit record directly from a keyboard.

The author would like to acknowledge the valuable assistance of Portia Isaacson, Ph.D., for her critique of this article leading to revisions of the original text.

Implementing Random Access for a Name and Address Retrieval System

By Gary Young

Many of the floppy disks and versions of BASIC support random access capability. Very little application software published so far uses this feature, however, partly because it is considered an advanced technique. Here is a name and address retrieval system that uses this capability. The system consists of two programs, written in North Star BASIC, to randomly add, delete, change, or search for data and sequentially read the data later.

THEORY

The disk used is the North Star Minifloppy. It is organized into 35 tracks, each track containing 10 sectors, and each sector containing 256 bytes. More simply, the disk can be thought of as 35 concentric circles each divided into ten 256-byte records. Random access in a hardware sense is the ability to read any one of these sectors given the track and sector number. Random access in a software sense is the ability to read any record within a file directly without reading all the preceeding records (sequentially). This means that in a 100 record file, reading only the 100th record would take 1/100th the time that it would take to read the file sequentially! For the North Star disk this is done by knowing the position of the first byte of a record relative to the beginning of the file.

A random access file is organized in some order based on a unique identifier for each record. This could be the account number, part number, or as used in this example, a person's name. This identifier is referred to as the "key" to a particular record. On large computer systems, this key is processed by a randomizing algorithm which then determines the position of the record within the file. For this example, the key is stored in a key list and the locations of the record within the file is determined by the location of the key within the list.

Some considerations must be kept in mind. How many records will be kept in the file, and how long will the key be? This will determine the size of the key list which will be the first record on the file. For the North Star disk, if the record is 255 bytes or less, two bytes are added to each record. If the record is over 255 bytes long, three bytes are added. The record locator is indicated in the program by "%VARIABLE" in the read and write statements. The word "NOENDMARK" prevents an end-of-file mark from being written.

In this personnel name and address retrieval system, the key is the name and is 25 bytes long. The file can contain 60 records, so the key list will be 1500 bytes long. A count of the number of entries on file is the first 5 bytes. The key list will be the first record on the file starting in position 6 for 1503 bytes. Each record contains 100 bytes plus 2. The starting position of the Nth record, then, is $5 + 1503 + (N-1)*102 + 1$.

THE NAME AND ADDRESS PROGRAMS

The first program will add, delete, change, or search for data. The record contains the address, city, state, zip code, area code and phone number, date correspondence sent, date correspondence received, and 35 characters of other data such as birthdate, etc. This other data will not be listed unless a "pseudo security" indicator was set previously. This feature prevents someone else from having access to the data unless they are smart enough to figure out the key from a listing of the program.

The program will type "REQ?" to request an action — A for adding a person to the file, D for deleting someone, C for changing some of the data, and S for search and list all the data for someone. The code exists to initialize the system by writing blank records and keys, but the check for the initialize request has been removed to prevent inadvertantly erasing the file.

On an ADD command, the program will search the list of keys for a free entry, then request all the data, one item at a time, storing the data in the record, and storing the name in the key list. Both the key list and record will then be written to the disk.

For a delete, change, or search, the program will request the name to search for. Any part of the name can be given and when a match occurs, the whole name will be printed back to be sure that it has found the correct one. For instance, if there are several "Bob"'s on the file and the name you enter is just "Bob", each "Bob" will be printed and you will be asked if that is the one you wanted. If the name cannot be found, the name will be shortened and searched again for a match.

On a DELETE request, the name is overwritten with an asterisk in the key list, and that entry is available for another add.

On a SEARCH request, the information on file is found, formatted, and printed.

When doing a CHANGE request, the program will ask which data to change, AD - address, CY - city, SZ - state and zip code, AP - area code and phone number, CS - date correspondence sent, CR - date correspondence received, OT - other data, and NM - name. The old data will be printed and prompted for the new data. If NM is entered, the list of keys is altered and rewritten to the disk.

Any combination of operations can be done. If a carriage return is received without a request, the program will stop.

The second program sequentially reads the random file and prints the data. It will print either a phone list, address list, or list all information maintained on each person.

CONCLUSION

These two programs, although useful in themselves, are only an example of how random access can be used to get selected data rapidly. The programs can be easily modified for any application. Other disk systems and versions of BASIC may require some modification to code, but the technique should be similar.□

SAMPLE RUN 1

```
RUN

PERSONNEL INFORMATION RETRIEVAL
REQ? A
NAME? JOHN DOE                                ADD PERSON 1
ADDR? 1234 PARK PLACE
CITY? LOS ANGELES
ST ZIP? CA90099
AREACODE AND PHONE? 2139998888
CORR SENT? 112277
CORR RCVD? 223344
OTHER? HAS A GREEN CAR
REQ? A                                        ADD PERSON 2
NAME? JOHN BROWN
ADDR? 333 SUNSET ST
CITY? SANTA MONICA
ST ZIP? CA90077
AREACODE AND PHONE? 2133991111
CORR SENT?
CORR RCVD?
OTHER?
REQ? MARY WEATHER                             ERROR
INVALID REQUEST
REQ? A                                        ADD PERSON 3
NAME? MARY WEATHER
ADDR? ON A STREET
CITY? BIGCITY
ST ZIP? CA55555
AREACODE AND PHONE? 3337779199
CORR SENT? 112277
CORR RCVD? 103077
OTHER? CUTE!!!
REQ? S                                        SEARCH AND PRINT
NAME??? JOHN
JOHN DOE
OK? N
JOHN BROWN
OK? Y
ADDR: 333 SUNSET ST
CITY: SANTA MONICA      , CA 90077
PHONE: 213    399-1111
SENT:          RCVD:
REQ? C                                        CHANGE DATA
NAME??? MARY
MARY WEATHER
OK? Y
DATA?? XX
NM, AD, CY, SZ, AP, CS, CR, OT
DATA?? CY
CITY: BIGCITY
?VENICE
DATA?? SZ
ST ZIP: CA55555
?CA90033                                      NULL RETURN
DATA??                                        TERMINATES COMMAND
REQ? D                                        DELETE PERSON
NAME??? JOHN
JOHN DOE
OK? N
JOHN BROWN
OK? Y
DELETED: JOHN BROWN
REQ?                                          NULL RETURN
 4632                                         TERMINATES PROGRAM
READY
```

SAMPLE RUN 2

```
RUN

PERSONNEL INFORMATION REPORTING
REPORT: PHONE (P), ADDR (A), OR ALL (D)? P
NAME                        AREA   PHONE #
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
JOHN DOE                           999-8888   11/22/77   22/33/44
MARY WEATHER                333    777-9199   11/22/77   10/30/77
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
READY
```

PROGRAM LISTING 1

```
1000 REM PERSONNEL NAME AND ADDRESS PROGRAM
1100 REM WRITTEN BY GARY YOUNG
1200 REM
1300 REM
1400 REM
1500 REM
1600 REM
1700 PRINT "PERSONNEL INFORMATION RETRIEVAL"
1800 DIM K9$(1500),R8$(100),K8$(35),A1$(1)
1900 REM J9 IS THE NUMBER OF RECORDS ON FILE
2000 J9=60
2100 REM R9 IS THE FILE OPEN AND KEYLIST LOADED FLAG
2200 R9=0
2300 REM S9 IS THE SECURITY INDICATOR
2400 S9=0
2500 REM S8 IS THE FLAG TO REWRITE THE KEYLIST AND ENTRY COUNT
2600 S8=0
2700 INPUT "REQ? ",A1$
2800 IF LEN(A1$)=0 THEN 22200
2900 IF A1$="A" THEN 5000
3000 IF A1$="D" THEN 14700
3100 IF A1$="C" THEN 15800
3200 IF A1$="S" THEN 13500
3300 IF A1$="Z" THEN S9=1
3400 IF A1$="Z" THEN 2700
3500 REM IF A1$="I" THEN INITIALIZE
3600 PRINT "INVALID REQUEST"
3700 GOTO 2700
3800 REM
3900 REM INITIALIZE THE SYSTEM
4000 K9=0
4100 R9=1
4200 OPEN #1,"PERSDATA"
4300 WRITE #1,K9,K9$
4400 FOR J=1 TO J9
4500 WRITE #1,R8$
4600 NEXT
4700 PRINT "FREE SPACE: ",FREE(0)
4800 GOTO 2700
4900 REM
5000 REM ADD AN ENTRY
5100 IF R9=0 THEN GOSUB 9300
5200 K=0
5300 K=K+1
5400 IF K>K9 THEN 5900
5500 K7=(K-1)*25+1
5600 REM AN OLD DELETED ENTRY IS AVAILABLE
5700 IF K9$(K7,K7)="*" THEN 6400
5800 GOTO 5300
5900 IF K<J9+1 THEN 6300
6000 PRINT "LIMIT EXCEEDED"
6100 GOTO 2700
6200 REM NEXT ENTRY IS AVAILABLE
6300 K9=K9+1
6400 K7=(K-1)*25+1
6500 REM CLEAN THE KEY LIST ENTRY
6600 K9$(K7,K7+24)="                         "
6700 INPUT "NAME? ",K8$
6800 K9$(K7,K7+24)=K8$
6900 R8$=""
7000 REM CLEAN THE RECORD
7100 FOR J=1 TO 10
7200 R8$=R8$+"          "
7300 NEXT
7400 INPUT "ADDR? ",K8$
7500 R8$(1,20)=K8$
7600 INPUT "CITY? ",K8$
7700 R8$(21,36)=K8$
7800 INPUT "ST ZIP? ",K8$
7900 R8$(37,43)=K8$
8000 INPUT "AREACODE AND PHONE? ",K8$
8100 R8$(44,53)=K8$
8200 INPUT "CORR SENT? ",K8$
8300 R8$(54,59)=K8$
8400 INPUT "CORR RCVD? ",K8$
8500 R8$(60,65)=K8$
8600 INPUT "OTHER? ",K8$
8700 R8$(66,100)=K8$
8800 K7=1508+(K-1)*102
8900 WRITE #1%K7,R8$,NOENDMARK
9000 S8=1
9100 GOTO 2700
9200 REM
9300 REM OPEN AND INITIALIZE
9400 OPEN #1,"PERSDATA"
9500 READ #1,K9,K9$
9600 R9=1
9700 RETURN
9800 REM
9900 REM SCAN THE KEYS
10000 S7=0
10100 INPUT "NAME??? ",K8$
10200 N1=LEN(K8$)
10300 P1=0
10400 IF N1<3 THEN 12200
10500 N2=0
10600 N2=N2+1
10700 IF N2>K9 THEN 11200
10800 N3=(N2-1)*25+1
10900 N4=N3+N1-1
11000 IF K9$(N3,N4)=K8$(1,N1) THEN 12400
11100 GOTO 10600
11200 IF P1<>0 THEN 11800
11300 INPUT "NO MATCH - RETRY (R) OR CONT (C)? ",A2$
11400 IF LEN(A2$)=0 THEN 11800
11500 IF A2$="R" THEN 10100
11600 IF A2$="C" THEN 11800
11700 GOTO 11300
11800 P1=P1+1
```

```
11900 N1=N1-5
12000 GOTO 10400
12100 PRINT "PERSON NOT ON FILE"
12200 P1=0
12300 RETURN
12400 PRINT K9$(N3,N3+24)
12500 INPUT "OK? ",A2$
12600 IF LEN (A2$)=0 THEN 13000
12700 IF A2$="Y" THEN 13000
12800 IF A2$="N" THEN 10600
12900 GOTO 12500
13000 P1=-N2
13100 K7=1508+(N2-1)*102
13200 READ #1%K7,R8$
13300 RETURN
13400 REM
13500 REM SCAN
13600 IF R9=0 THEN GOSUB 9300
13700 GOSUB 9900
13800 IF P1<0 THEN 13900 ELSE 2700
13900 PRINT "ADDR: ",R8$(1,20)
14000 PRINT "CITY: ",R8$(21,36)",",R8$(37,38)," ",R8$(39,43)
14100 PRINT "PHONE: ",R8$(44,46)," ",R8$(47,49),
14200 PRINT "-",R8$(50,53)
14300 PRINT "SENT: ",R8$(54,59),"  RCVD: ",R8$(60,65)
14400 IF S9<>0 THEN PRINT "OTHER: ",R8$(66)
14500 GOTO 2700
14600 REM
14700 REM DELETE
14800 IF R9=0 THEN GOSUB 9300
14900 GOSUB 9900
15000 IF P1<0 THEN 15100 ELSE 2700
15100 N2=-P1
15200 N3=(N2-1)*25+1
15300 PRINT "DELETED: ",K9$(N3,N3+24)
15400 K9$(N3,N3)="*"
15500 S8=1
15600 GOTO 2700
15700 REM
15800 REM CHANGE
15900 IF R9=0 THEN GOSUB 9300
16000 GOSUB 9900
16100 IF P1<0 THEN 16200 ELSE 2700
16200 INPUT "DATA?? ",A2$
16300 IF LEN(A2$)=0 THEN 21600
16400 IF A2$="NM" THEN 20900
16500 IF A2$="AD" THEN 17400
16600 IF A2$="CY" THEN 17900
16700 IF A2$="SZ" THEN 18400
16800 IF A2$="AP" THEN 18900
16900 IF A2$="CS" THEN 19400
17000 IF A2$="CR" THEN 19900
17100 IF A2$="OT" THEN 20400
17200 PRINT "NM, AD, CY, SZ, AP, CS, CR, OT"
17300 GOTO 16200
17400 PRINT "ADDR: ",R8$(1,20)
17500 INPUT K8$
17600 R8$(1,20)="                    "
17700 R8$(1,20)=K8$
17800 GOTO 16200
17900 PRINT "CITY: ",R8$(21,36)
18000 INPUT K8$
18100 R8$(21,36)="                "
18200 R8$(21,36)=K8$
18300 GOTO 16200
18400 PRINT "ST ZIP: ",R8$(37,43)
18500 INPUT K8$
18600 R8$(37,43)="       "
18700 R8$(37,43)=K8$
18800 GOTO 16200
18900 PRINT "AC+PHONE: ",R8$(44,46)," ",R8$(47,53)
19000 INPUT K8$
19100 R8$(44,53)="          "
19200 R8$(44,53)=K8$
19300 GOTO 16200
19400 PRINT "SENT: ",R8$(54,59)
19500 INPUT K8$
19600 R8$(54,59)="      "
19700 R8$(54,59)=K8$
19800 GOTO 16200
19900 PRINT "RCVD: ",R8$(60,65)
20000 INPUT K8$
20100 R8$(60,65)="      "
20200 R8$(60,65)=K8$
20300 GOTO 16200
20400 PRINT "OTHER: ",R8$(66)
20500 INPUT K8$
20600 R8$(66,100)="                                   "
20700 R8$(66,100)=K8$
20800 GOTO 16200
20900 INPUT "NEW NAME? ",K8$
21000 N2=-P1
21100 N3=(N2-1)*25+1
21200 K9$(N3,N3+24)="                         "
21300 K9$(N3,N3+24)=K8$
21400 S8=1
21500 GOTO 16200
21600 N2=-P1
21700 K7=1508+(N2-1)*102
21800 WRITE #1%K7,R8$,NOENDMARK
21900 GOTO 2700
22000 REM
22100 REM TERMINATE
22200 PRINT FREE (0)
22300 IF S8=0 THEN 22500
22400 WRITE #1%0,K9,K9$,NOENDMARK
22500 CLOSE #1
22600 END
READY
```

PROGRAM LISTING 2

```
1000 REM PRINT THE PERSONNEL FILE OF NAME AND ADDRESSES
1100 REM WRITTEN BY GARY YOUNG
1200 REM
1300 PRINT "PERSONNEL INFORMATION REPORTING"
1400 DIM K9$(1500),R8$(100),A1$(1)
1500 OPEN #1,"PERSDATA"
1600 READ #1,K9,K9$
1700 INPUT "REPORT: PHONE (P), ADDR (A), OR ALL (D)? ",A1$
1800 IF A1$="P" THEN 4300
1900 IF A1$="A" THEN 2500
2000 REM S WILL DUMP ALL DATA INCLUDING OTHER INFORMATION
2100 IF A1$="S" THEN 2500
2200 IF A1$<>"D" THEN 1700
2300 REM
2400 REM ADDRESS OR DUMP ALL
2500 FOR J=1 TO K9
2600 READ #1,R8$
2700 N1=(J-1)*25+1
2800 IF K9$(N1,N1)="*" THEN 3900
2900 N2=N1+24
3000 GOSUB 6900
3100 PRINT K9$(N1,N2)
3200 PRINT R8$(1,20)
3300 PRINT R8$(21,36),", ",R8$(37,38)," ",R8$(39,43)
3400 IF A1$="A" THEN 3900
3500 PRINT R8$(44,46),"   ",R8$(47,49),"-",R8$(50,53)
3600 PRINT "SENT: ",R8$(54,59),"  RCVD: ",R8$(60,65)
3700 IF A1$<>"S" THEN 3900
3800 PRINT "OTHER: ",R8$(66,100)
3900 NEXT
4000 GOTO 7400
4100 REM
4200 REM PHONE LIST
4300 PRINT "NAME",
4400 PRINT TAB(27),"AREA   PHONE #",
4500 PRINT
4600 GOSUB 6900
4700 K=0
4800 FOR J=1 TO K9
4900 READ #1,R8$
5000 N1=(J-1)*25+1
5100 N2=N1+24
5200 IF K9$(N1,N1)="*" THEN 6400
5300 K=K+1
5400 PRINT K9$(N1,N2),
5500 IF R8$(44,46)="213" THEN 5700
5600 PRINT TAB(27),R8$(44,46),
5700 PRINT TAB(34),R8$(47,49),"-",R8$(50,53),
5800 IF R8$(54,59)="      " THEN 6000
5900 PRINT TAB(45),R8$(54,55),"/",R8$(56,57),"/",R8$(58,59),
6000 IF R8$(60,65)="      " THEN 6200
6100 PRINT TAB(56),R8$(60,61),"/",R8$(62,63),"/",R8$(64,65),
6200 PRINT
6300 IF INT(K/3)=K/3 THEN GOSUB 6900
6400 NEXT
6500 GOSUB 6900
6600 GOTO 7400
6700 REM
6800 REM PRINT SEPARATOR DASH LINE
6900 FOR D=1 TO 70 STEP 2
7000 PRINT TAB(D),"-",
7100 NEXT
7200 PRINT
7300 RETURN
7400 END
READY
```

The Automated

Audrey Roche had just received a telephone call.

"Can we interest you in using our computer service for your accounting business?" the voice had asked.

"No, thank you," Roche had replied. "I already have a computer."

What does her computer do? It prepares a chart of accounts for each of her 18 clients and provides them monthly with a complete set of journals, a thoroughly updated general ledger, a balance sheet, income statement and all supporting schedules.

In fact, it can do anything a computer service can do for an accountant, which is what keeps her competitive in an increasingly automated field. For this reason, Roche feels that the use of microcomputers by small accounting firms will increase.

"I don't think anybody's going to be doing accounting by hand once computers become widely available and more people see them used," she says. "Anybody in the accounting profession who graduated more than ten years ago and hasn't gone back and learned about computers may be frightened by them. But once they get a taste of it, and find out that it's really not that scary, they won't be doing anything by hand.

"I wouldn't do the simplest little account by hand because it would take longer to type out the income statement and balance sheet on the typewriter than it would to key the stuff in and get it printed out on the printer."

Indeed, one of the system's chief benefits is the visibility that is possible when the hard copy is produced. This has helped Roche's business immensely.

For example, in order to determine how much money was spent on advertising, the hand system would require going to the general ledger under advertising and then to the cash payments journal for the month's total. But one still wouldn't know to whom the money was paid or what it was for.

The computerized version, on the other hand, is more thorough and accessible. Roche explains: "You just go to the general ledger account called advertising and you can see every dime that was spent on it."

The general ledger package, which is written in BASIC, is the dominant feature of the system. It contains two major categories: Update Data Base, including chart of accounts, transactions, vendors and beginning balances, and Print Reports, consisting of chart of accounts, journals, registers, vendors, trial balance, income statement, balance sheet and general ledger.

There's also a line spacing control that designates the page format, and a general ledger code, which indicates whether the account is simply a heading, totalized account, general ledger, subsidiary ledger, total of subsidiary accounts that is printed in the general ledger, income statement account or transfer to the balance sheet of income for the year.

For each account, a beginning balance and year to date balance is recorded with a description of the account, usually in the form of a name. Then the account set number is programmed. This command covers such important categories as the balance sheet, income statement, directions for number of columns and information as to whether it should be printed at all or if it's a subsidiary account.

In addition to the general ledger program, Roche has a word processing package which she employs in an inventive and efficient fashion. First, she set up a format for W-2 income tax statements and purchased some self-feeding forms.

"When we zapped out W-2's like mad," she recalls proudly. "We did about 200 and it was very fast. But getting the program to work took longer than doing the W-2's because I was not familiar with it and fouled things up quite often."

THE SYSTEM

Roche's system is a COMPAL 80 microcomputer with 64K memory, a Micropolis Dual Disk, Texas Instruments Model 810 printer and a Hitachi video monitor.

The system's $11,000 price tag also includes word processing and general ledger programs, manuals, warranty service and programming classes for two people.

The Texas Instruments printer operates at 150 characters per second, is bi-directional and costs under $2,000.

The computer itself is a turn-key system, which, as the name implies, is operational as soon as the machine's turned on. Roche is especially excited about it because she can fit a full year of transactions (over 300 for one account) onto a single disk.

All of her accounting information is stored on floppy disks, two of which are used for each client: one for input and one for a backup copy. Each floppy disk costs about seven dollars and can hold about 325 bytes of information.

THE REAL WORLD

Roche says that she's had no problems with the computer so far, and that the only upkeep it required was a

Accountant

By Mathew Tekulsky

trip to Micropolis to fix the disk drive. And that was covered under the guarantee.

However, recognizing the difficulty involved in developing original software, she stresses the importance of purchasing hardware from people who know how to use it and are willing to work with their customers.

But although she's had her computer since November 1, 1977 and it was fully operational by January 1, 1978, Roche only has half of her clients on it so far.

"We're just really new at it," she admits, "and the software wasn't complete when we got the system. We're hoping to get it really running and it's getting better every day. But it takes time to set up each file, from designing the chart of accounts to being able to key in a program entry."

In order to keep the system operating smoothly, she employs the services of a programmer.

"Right now the programmer is trying to speed things up and make the sorts faster with little tricks and techniques," she reports. "But the system is working beautifully."

Has she considered doing her own programming?

"If you're a super programmer and you understand accounting well enough to do your own, you must recognize that it will probably take six months to write the program," she explains. "The general ledger's a long program and it's much more complicated than most people think."

The reason for this is that accounting is much more complicated than most people think.

"Accounting is not just adding and subtracting," she goes on. "There are a lot of complications involved in it. And you need that software. If you're a large company and you have a group of programmers and a basic outline of an accounting system, you could do it yourself.

"But any small person who thinks he can sit down and write his own programs and be cost effective is fooling himself. You're talking about $1,500 for the software. Just divide that by six months of your time and you'll find it couldn't possibly be worth it."

One thing's for sure, however. According to Roche, her computer is certainly worth it. In fact, she expects it to pay for itself in just two years.

THE FUTURE

In the future, Roche plans to expand her system to include more hardware and software. "My hope is to go to a third disk drive, put the general ledger on it and keep adding to it all year, erasing all the data on a regular disk as I finish each month," she says.

"We've talked to our programmer about it," she continues, "and although it seems very feasible to me, I'm not sure we have totally convinced him. But we're working on it."

The purpose of having three disks is essentially to save time. When November's transactions are called up now, for instance, the previous ten months are printed individually for the year to date. This takes a while. But with three disks, the time required would be smaller and the general ledger would be continually updated. If an

error were made, a correcting entry could be inserted, just like in a hand system.

"I think you can find most of those types of errors in advance," Roche contends, "so it's not the end of the world. And it would save a lot of time."

She's also intending to expand her software into payroll accumulation, which would give her quarterly returns and W-2's at the end of the year. And by inputting data throughout the year and calling it up at year's end, she'll be able to do both hers and her client's taxes.

In addition, Roche is looking forward to attaching her word processor to an IBM typewriter in the office.

"I'm not getting much encouragement from people on our ability to adapt that to our system," she reports, "although people have done it. There are converters out that cost much more than the typewriters cost. But that's one of the things I really want to get. Then we'll have a word processing capability that can do business letters and save secretarial time."

Her other plans include using an investments analysis program.

"If a client wants to invest in a property," Roche explains, "they can give us the pertinent information and we can run a schedule off that shows what their expected cash flow, expected return of investment and tax consequences of such a move would be. We'll probably expand it into other things, too."

How long will these expansions take?

"About a year," she says. "I'm still trying to get people on the computer, much less trying anything new. And when we have a chance, we also definitely want to have some programs of our own to develop."

In the meantime, she's satisfied with the ability to have a trial balance in about 70 seconds and to run off all her reports in about an hour. This would no doubt be impossible to do manually.☐

Complete Data Base

By Peter Reece

INTRODUCTION

One of the most common uses of a computer is the manipulation of large amounts of data in a utilitarian task-determined manner. That is, by selective manipulation or scanning of knowledge bases, the computer can yield rapid summary information which is representative of the complete data base. This manipulation, commonly known as data base management, is unfortunately usually relegated to the large computer only. The small system user who wishes to organize office inventory, book lists, mailing labels, and the like, is usually left out in the cold.

The IDMAS (Interactive Data Base Manipulator And Summarizer) system is a remedy to the problems of the small user who wishes large system features. IDMAS allows the user to selectively scan, summarize, total, count, change, enter, delete, and encode data from a data base without any knowledge of the internal workings of the computer or program. Commands are interpreted through an English language parser which can be modified or augmented very easily to enable the user to utilize the English language subset he prefers. There may be as many data bases on the system as the user's facility allows. In addition, while the program is included here in direct access disk mode (available in most microfloppy software) modification to a sequential access file structure is straightforward.

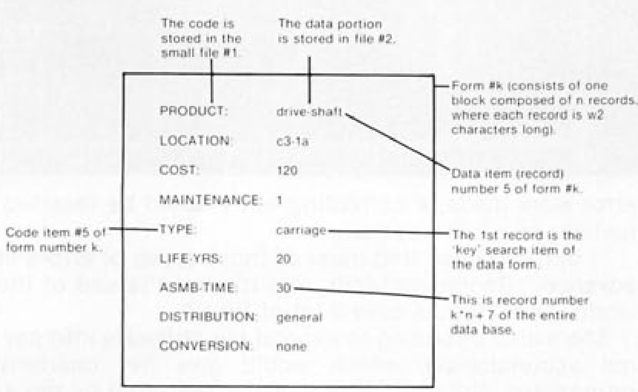


Figure 1. A small annotated sample form from an assembly line data base.

IDMAS is written in PDP-10 BASIC — a BASIC dialect which is widely accepted (see IDMAS listing). If the error message and ASCII code in the program is minimized for a given application, the entire source will fit into about 12K of core on the average small computer. Further, since the various commands are modular, they may be removed at will if not required in a given application, in order to drain core even less.

FILE STRUCTURE

There are two data files for every data base used by IDMAS. The first is a small file containing the codes or 'names' assigned by the user to the various items in his data base. The second file contains the actual data which corresponds to these names. This is the larger of the two files. For example, suppose that the user is concerned with the length of time it takes to assemble some automotive components and wishes to store this information in a data base. As Figure 1 shows, the time, say 30 minutes, would be stored as data in file two while the name of that data — asmb-time — would be stored in file one.

Each item of data and its name are stored in a single record in their respective files. These records are always the same distance from the start of this block of data. Thus, in our example, asmb-time is stored at the seventh record of the block, and '30' is stored at the seventh record of the block of file two.

A block consists of all records which correspond to a given item in the data which is of particular importance. This is the 'key' item, and is usually the main item of interest in the data base. If we think of a data base consisting of mailing lists, the key item might be postal zone since it is the most general designation. The next most important item might be state, then county, city, street, and so on. The key item would be the first item in the hierarchy. All records following the key will in some way be tied to the key. (Note that IDMAS does not require the key item be the main item in a block, but from a user standpoint, and simply by convention, one item is usually designated as the key.)

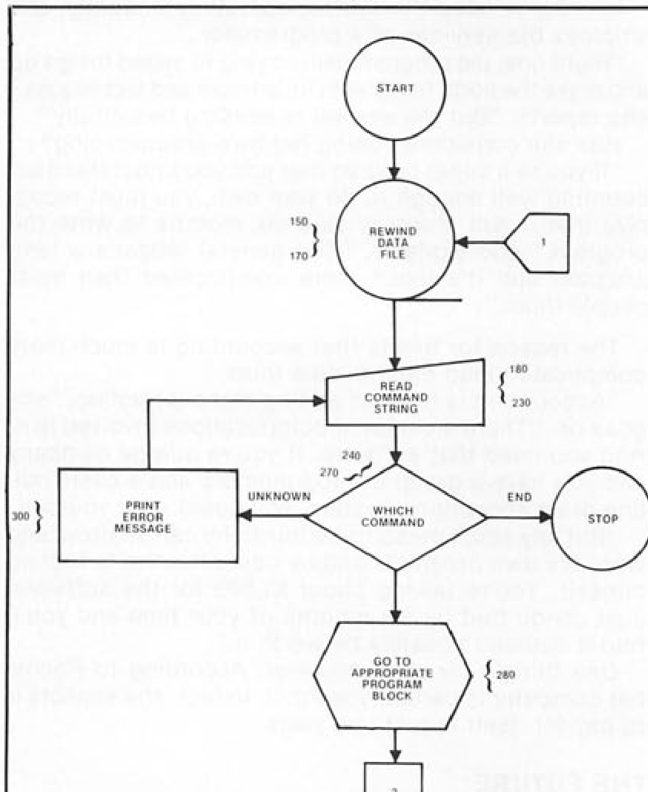


Figure 2. The initializing loop of the program (Numbers in single brackets indicate line numbers in the program.)

A 'form' consists of the key plus its related records. There may be one and only one code form per data base in IDMAS. This is because all data in file two is assigned names through the code form of file one. More code forms would lead to confusion, and the program automatically prevents the occurrence of two code forms. The number of different codes per data base, however, is unlimited. A 'form', then, relates to the total structure enclosed in the box in Figure 1, while a 'block' refers to the physical grouping of the records in the data file.

(That is, 'block' takes on the common meaning of data block on a disk or tape file.) The user may choose the length of each record within a block by adjusting the parameter 'w2'. Hence, if w2 = 80, each record in file two, the data file, will be 80 characters in length. (Blanks are added if all 80 are not used.)

In performing a search, the program computes the length of a form from the number of names in the list of codes in file one and adds to this the distance of the user selected record from the key item. For example, to search the data base for all assembly times, as in Figure 1, the program would read records 7, 16, 25, etc. In this way, only records relevant to the search are read, thereby saving considerable read time.

THE SEARCH

The actual mechanics of this search is illustrated in flow form in Figure 3. After deciding that a search has been requested (see Figure 2) the program reads the code table to compute the validity of the names requested by the user input string. Such an input string is illustrated in the example in the first three lines of Figure 6.

Once it has been determined that the input string contains valid requests, the search is begun. Each search item of the input string is compared to the information residing in the appropriate record as computed by the method already outlined. For example, as illustrated in Figure 6, three records would be read per form, and the information in the seventh record, for example, would be checked to see if it is greater than fifteen. Totals, detailed counts, etc. (see command list outlined below) would be performed once it had been determined that all of the requested information in this form had matched the user's input specifications (e.g., that there was not a cost of 10 in the form of the example in Figure 6).

In this manner, the search would continue until each form in the data file had been read. A summary according to the user's request and previous commands would then be printed, and the program would await the next command.

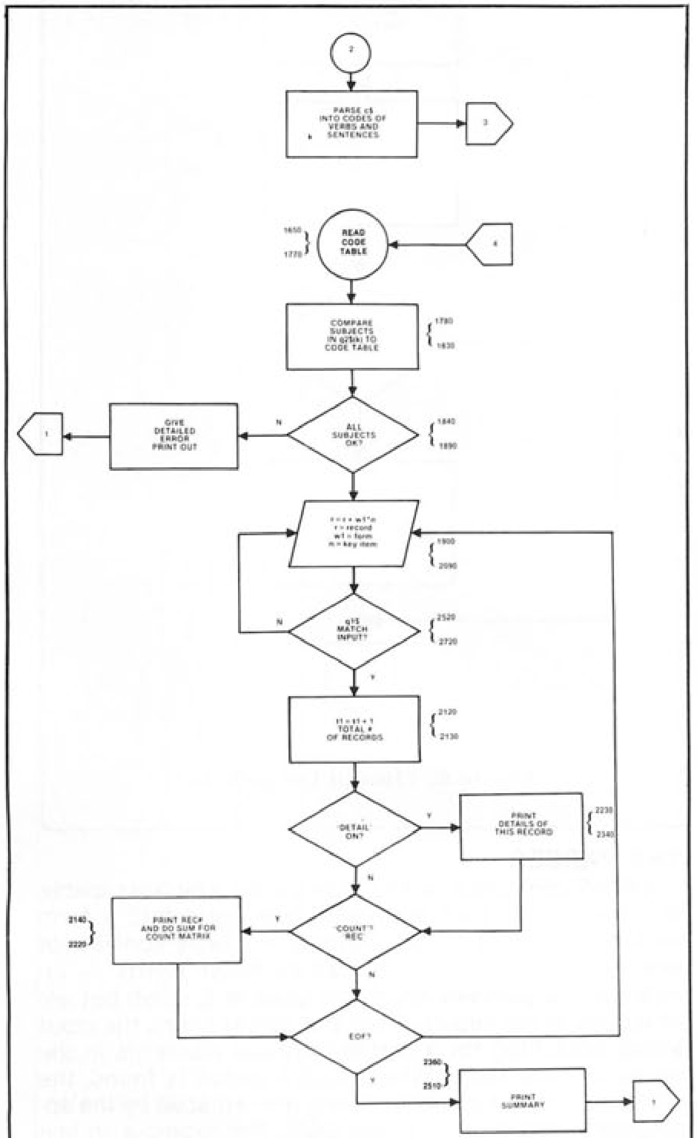


Figure 3. The search loop.

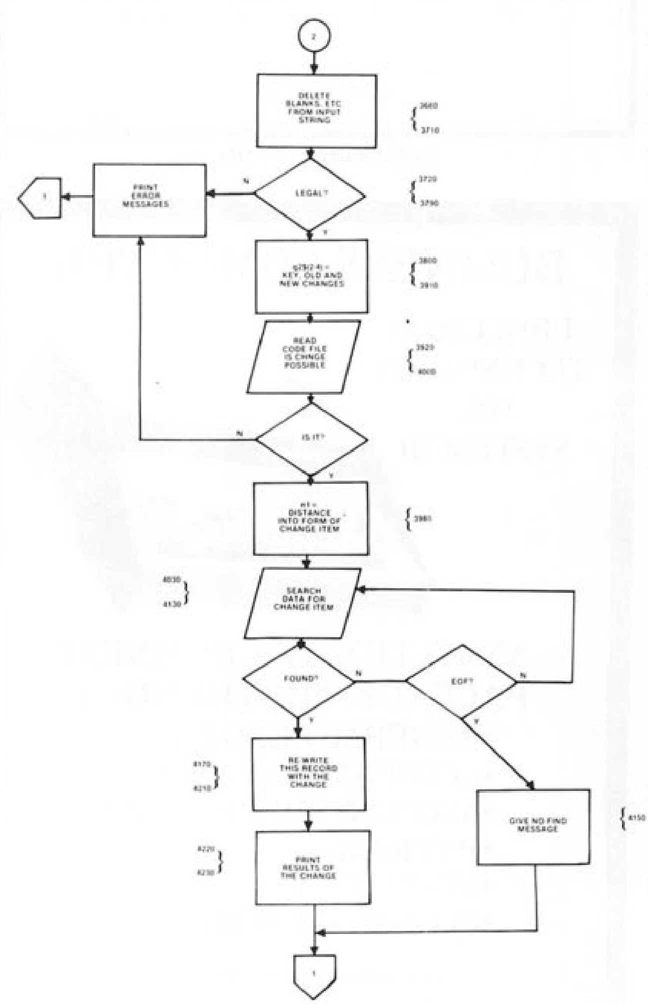


Figure 4. The 'change' command.

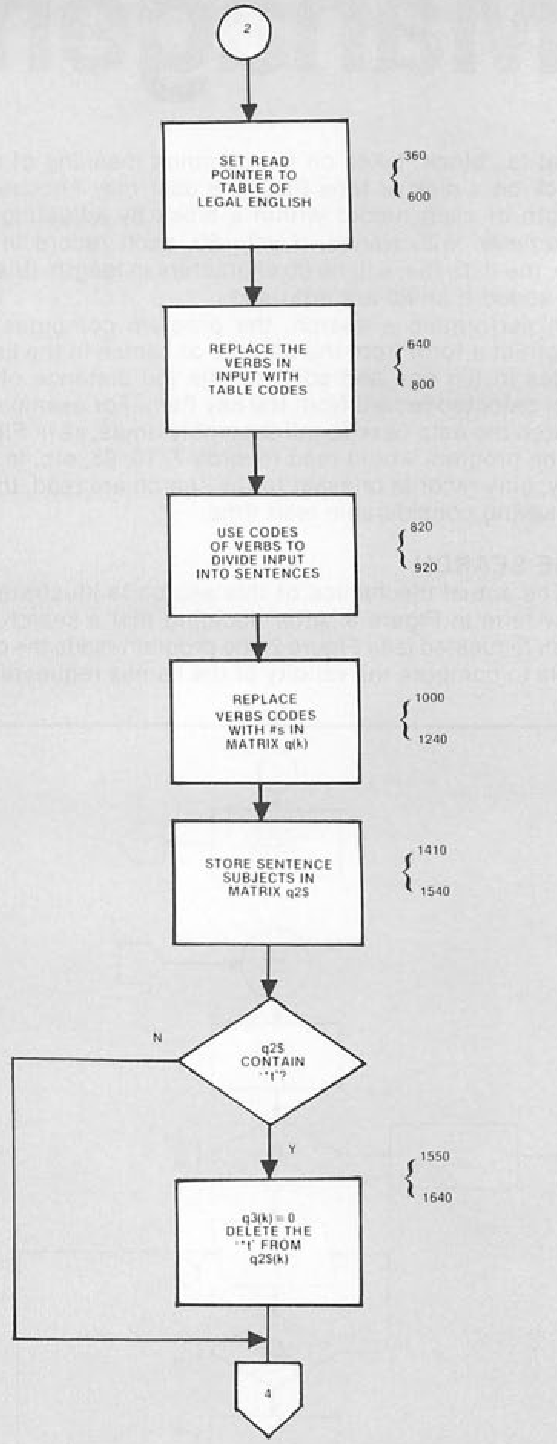


Figure 5. Flow of the parser.

THE PARSER

IDMAS contains a table driven parser which is capable of encoding English language user input into a form usable by the rest of the program. The table consists of verbs and verb phrases, as well as 'noise words' — articles and adjectives which are used in English but are of no use in the search. First the parser scans the input string searching for a match between elements in the string and the table. When such a match is found, the matched characters in the string are replaced by the appropriate verb code from the table. For example, in line 510 of the program, it can be seen that the phrase 'is not' is replaced by the code '.not.'. That is, the table is

composed of pairs — the first word in the pair is the match item, the second is the replacement item. If the replacement item is a '9', the parser automatically replaces the match item in the input string with blanks.

Searches are performed...by treating all the words between the 'ands' as separate sentences. A search is successful if. . .a given form meets the conditions specified by each sentence in the input string.

Note that the table may contain any verbs or phrases which the user deems appropriate to his task. Hence, the input string has considerable flexibility. The table may also be as long or short as the user wishes.

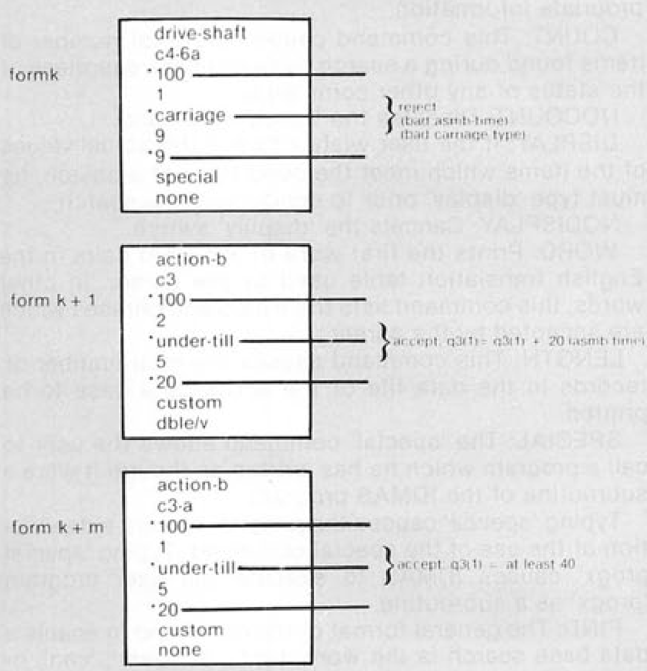
The parser's next step is to remove the blanks from the input string. At this point, the input string will consist only of subjects, verbs, objects, and connectors. For example, if the input string is:

FIND AN ASMB-TIME WHICH IS MORE THAN 15*t AND WHOSE TYPE IS NOT A CARRIAGE AND WHICH SHOULD HAVE A COST OF 100.

then at this point in the parsing procedure, the input string would appear as:

ASMB-TIME.GT.15*tANDTYPE.NOT.CARRIAGEANDCOST.IS.100

FIND AN ASMB-TIME WHICH IS MORE THAN 15*t AND WHOSE TYPE IS NOT A CARRIAGE AND WHICH SHOULD HAVE A COST OF 100.



*The program computes which records are of use and reads *only* those records — thereby speeding search time considerably.

The minimum output (i.e. if no additional forms where accepted, and no special program switches where in effect) would be:

I HAVE FOUND 2 ITEMS.
TOTAL ASMB-TIME = 40.

Figure 6. Example of a simple search through data.

The parser now creates a number of matrices which categorize the information in the above string. First the matrix q$(k) is created, where each 'k' contains one of the sentences in the input string. The matrix q1(k) is then created and consists of numbers which represent the verbs of the input string. The objects of each sentence — that is, the sought items in the data — are contained in the matrix q1$(k). Finally, q2$(k) is loaded with the subjects of the sentences. To illustrate all of this, using the above example, we might have the following:

	q1(k)	q1$(k)	q2$(k)	q$(k)
k = 1	3	15	asmb-time	asmb-time.gt.15
k = 2	2	carriage	type	type.not.carriage
k = 3	1	100	cost	cost.is.100

The final task of the parser is to assign a subject to sentences which lack one. This is done by assuming that the subject of the previous sentence of the input string also applies to the present sentence. If no previous message existed, an error message is generated.

At this time too, the matrix q3(k) is loaded with a one or a zero per k according to whether or not a total is desired for objects in q1$(k). For example, in the input string illustrated above, q3(1) = 0, q3(2) = 1, q3(3) = 1, since a total of assembly times only was requested.

The general flow of the parser, as well as the program line numbers which correspond to the various tasks, is illustrated in Figure 5.

COMMANDS

Below is a listing of command words accepted by IDMAS, with a description of each. Each command may be abbreviated to the first three letters if desired.

HELP: prints a list of commands.

DELETE: Deletes a particular form from the data base. Deletions are always by key item. For example, using the illustration in Figure 6, typing 'DELETE ACTION-B' would delete form number k + 1. To delete form k + m, 'DELETE ACTION-B' would have to be typed again. This is a safety feature in the event that duplicate key items exist (it is up to the user whether or not his key items are unique).

Suppose that form k + 1 only had been deleted. The next time that the 'ADD' command was issued by the user (see below) the space left empty by form k + 1 would be filled. In this way IDMAS prevents the existence of holes in the data base, minimizing storage cost.

ADD: To add a form to the data base, the user need type only the word 'add' plus the name of the key item plus the data for the key item. For example, to add the product 'mud' to the data base, type 'ADD PRODUCT MUD'. IDMAS would then prompt the user for the rest of the items in the form, e.g. the value of 'location', 'cost', and so on.

A switch exists in the program (see 'add' subroutine) which allows the user to have multiple key items of identical value should he wish to do so. Otherwise the program will automatically produce an error message if the user attempts to create forms with non-unique keys. This feature therefore allows maximum versatility in the use of non- or pure hierarchical data base form structures.

DONE: This command will terminate the action of any other command which is waiting for input. For example, if half way through 'add'ing a form the user decides that he doesn't wish to 'add' this form after all, typing 'done' will terminate the add command and return the user to his initial state prior to the 'add' command.

CREATE: To create a new data base, simply type 'create'. IDMAS will then prompt for names of code items and internally assign files one and two for future use by this data base.

CODE: This command allows the user to scan raw data as it actually exists in his data file by naming the specific records he wishes to see. For example, if the user wishes to scan the contents of records 200-204, he need simply type 'code'. IDMAS will then prompt for the first and last records desired (200 and 204, respectively), then type the contents of these records in their actual stored state.

RECORD: Typing 'record' activates a program switch which causes the record numbers of all key items activated during a search (assuming the form matches the search specifications) to be printed.

NORECORD: Disables the 'record' switch.

SHOW: This causes the chosen form to be printed as in Figure 1. For example, to print the form whose key item is 'mud', type 'SHOW PRODUCT MUD'.

CHANGE: Figure 4 illustrates the flow of the change command. Suppose, for example, that the user wished to change the 'type' in Figure 1 from 'carriage' to 'unit/a'. To do this he need simply type 'CHANGE TYPE IN PRODUCT = DRIVE—SHAFT TO UNIT/A'. The program would then search the data base for the key item named, then perform the change. If the search and change were successful, IDMAS would type:

'TYPE CHANGED FROM CARRIAGE TO UNIT/A IN PRODUCT DRIVE-SHAFT'.

TOTAL: This enables the facility whereby user designated *numeric* items may be summed together during a search (assuming the conditions of a search are met). The user specifies which items he wishes IDMAS to sum by adding '*t' following the object(s) of a sentence or sentences. Figure 6 illustrates an example of the use of the 'total' command. As many objects as the user wishes may be totaled per input string.

REASSIGN: This allows the user to end (i.e., close) the data base he is presently working with and proceed to a different data base. IDMAS prompts the user for the appropriate information.

COUNT: This command causes the total number of items found during a search to be printed, regardless of the status of any other commands.

NOCOUNT: Disables the 'count' command.

DISPLAY: If the user wishes to see the actual values of the items which meet the conditions of a search, he must type 'display' prior to conducting the search.

NODISPLAY: Cancels the 'display' switch.

WORD: Prints the first word of the word pairs in the English translation table used by the parser. In other words, this command lists the words and phrases which are accepted by the parser.

LENGTH: This command causes the total number of records in the data file of the active data base to be printed.

SPECIAL: The 'special' command allows the user to call a program which he has written as though it were a subroutine of the IDMAS program.

Typing 'special' causes the program to print a description of the use of the special command. Typing 'special progx' causes IDMAS to execute the user program 'progx' as a subroutine.

FIND: The general format of the command to enable a data base search is the word 'find', 'suggest', 'can', or 'match' followed by sentences joined by the Boolian connective 'and'.

A sample format might appear as follows:

FIND subject1 verb object1 AND article subject2 verb phrase object2. Or, using the form of Figure 1:

FIND type is carriage AND a cost which isn't 120.

Searches are performed, as explained when the parser was discussed above, by treating all the words between the 'ands' as separate sentences. A search is successful if and only if a given form meets the condi-

tions specified by *each* sentence in the input string. In other words, the 'AND' is a logical, or Boolian 'and'.

All searches will print as a minimum the total number of forms that met all of the conditions of the search, if other switches have not been set.

CONCLUSION

The appendix gives several examples of the use of IDMAS with a very simple data base. It is hoped that by adapting IDMAS to your system, and taking advantage of its versatility and flexibility you will find that the task of data manipulation becomes easier, more useful and enjoyable.□

PROGRAM LISTING

```
00010REM *********************************************************
00012REM *                                                       *
00014REM *        IDMAS: A DATA BASE MANAGEMENT SYSTEM           *
00016REM *                                                       *
00040REM *              by Peter Reece                           *
00042REM *                                                       *
00060REM *********************************************************
00070dim q$(20),q1(20),q1$(20)
00080dim q2$(20),q2(20)
00090dim q3(20)
00110goto530
00120REM
00130REM ***** 'w2' is the number of columns allowed for a search item ****
00140w2=20
00150print" "
00152rem rewind data files
00160restore:1
00170restore:2
00180print "Ready: ";
00182rem read in a command
00190input c$
00200if c$="end"then1470
00210n=instr("helcoddisnodendexitswrecsortfincounoctotnotspe",left$(c$,3))
00220if n=0then250
00230n=int(n/3)+1
00240onnthen2950,3480,4250,4280,3383,3393,323,5070,5100,350,4660,1690,5280,320,4710
00250n=instr("natmacsubsis cancresubnsubsnoworlenass",left$(c$,3))
00260if n=0then290
00262rem choose a subsection, then go there
00270n=int(n/3)+1
00280onnthen350,350,350,350,350,2730,3373,3673,3670,4300,5160,5120,9000
00290gosub3440
00300print"*** I do not recognise your command ***"
00310goto150
00320gosub3440
00322rem reach here if new commands are being debugged
00330print"Command not yet implimented."
00340goto150
00350REM
00360REM ****************** FIND COMMAND ******************
00370REM first formalise verbs and eliminate noise words
00380data "is it possible to have","?"
00390data"is there","?","can there","?"
00400data "find","?","make","?","match","?","is it possible","?"
00410data "can","?","?","?"
00420data"at",".and."
00430data "who's",".is.","whose","?"
00440data "and","#","was not",".not."
00450data "should have had",".is.","should have",".is."
00460data "of",".is.","had",".is.","has",".is.","have",".is."
00470data "for",".is.","was",".is.","will be",".is.","should be",".is."
00480data "if","?","which","?","that","?","who","?","the","?"
00490data "is greater than",".gt.","is less than",".lt."
00500data "a","?","an","?","in","?","on","?"
00510data "are",".is.","is not",".not.","are not",".not."
00520data "with","and","<",".lt.",">",".gt.","isn't",".not."
00530data "wasn't",".not.","haven't",".not.","shouldn't",".not."
00540data "is",".is.","&","and","+","and"
00550data "with",".and.","to","?"
00560data "should","?","be",".is.","yet","?","is more than",".gt."
00570data "???","?"
00580REM ............... parser begins here ..................
00590c7=0
00600restore
00610n=instr(c$,"?")
00620if n=0then640
00630c7=444
00640read a$,b$
00650if a$="???"then810
00660n=instr(c$,a$)
00670if n=0then640
00680l=len(a$)
00690n1=n+1
00700if mid$(c$,n1,1)=" "then 730
00710n=instr(n1,c$,a$)
00720goto670
00730if n=1then760
00740c$=mid$(c$,1+2)
00750goto660
00760if b$="?"then790
00770c$=mid$(c$,1,n-1)+b$+mid$(c$,n+l)
00780goto660
00790c$=mid$(c$,1,n-1)+mid$(c$,n+l)
00800goto660
00810REM
00820REM  Now isolate the individual sentances into q$(*)
00830i=0
00840i=i+1
00850k$="#"
00860l=len(k$)
00870n=instr(c$,k$)
00880if n=0then920
00890q$(i)=mid$(c$,1,n-1)
00900c$=mid$(c$,n+l)
00910goto840
00920q$(i)=c$
00930REM q$(*) now contains the sentances to be 'anded'
00940REM Now eliminate all blanks from the sentance strings.
00950fork=1toi
00960c$=q$(k)
00970gosub1290
00980q$(k)=c$
00990nextk
01000REM  now isolate the verbs
01010matq1=zer
01020fork=1toi
01030n=instr(q$(k),".is.")
01040if n=0then1080
01050q1(k)=1
01060q1$(k)=mid$(q$(k),n+4)
01070goto1240
01080n=instr(q$(k),".not.")
01090if n=0then1130
01100q1(k)=2
01110q1$(k)=mid$(q$(k),n+5)
01120goto1240
01130n=instr(q$(k),".gt.")
01140if n=0then1170
01150q1(k)=3
01160goto1060
01170n=instr(q$(k),".lt.")
01180if n=0then1210
01190q1(k)=4
01200goto1060
01210gosub3440
01220print"** I do not recognize the verb in '";q$(k);"' **"
01230goto120
01240nextk
01250REM
01260REM q$(*)=sentances, q1(*)=verb codes,  q1$(*)=items saught
01270REM i=# of sentances.
01280goto1410
01290REM Routine to eliminate blanks from the enter string 'c$'
01300i1=0
01310l=len(c$)
01320i1=i1+1
01330if i1>lthen1400
01340if mid$(c$,i1,1)<>" "then1320
01350if i1<>1then1380
01360c$=mid$(c$,2)
01370goto1310
01380c$=mid$(c$,1,i1-1)+mid$(c$,i1+1)
01390goto1310
01400return
01410REM Load q2$(*) with sentance subjects for further messages.
01420REM Note that if a sentance lacks a subject then it is assumed
01430REM that the verb refers to the subject of the previous sentance.
01440fork=1toi
01450n=instr(q$(k),".")
01460if n=0then1210
01470if n>1then1510
01480gosub3440
01490print"** There is no subject in the sentance '";q$(k);"' **"
01500goto150
01510q2$(k)=mid$(q$(k),1,n-1)
01520if q2$(k)<>" "then1540
01530q2$(k)=q2$(k-1)
01540nextk
01550REM search subjects for 'total' commands
01560q3=-1
01570c8=0
01580fork=1toi
01590n=instr(q$(k),"#t")
01600if n=0then1640
01610q2$(k)=mid$(q2$(k),1,n-1)
01620q3(k)=0
01630c8=444
01640nextk
01650REM ............................. Parser ends here .....................
01660REM Now search a code table for matches to the subjects in q2$(*)
01670REM and store the record addresses.
01680set2:1
01690input:2,c$
01700w1=val(c$)
01710k=0
01720k=k+1
01730if k>ithen1890
01740n=0
01750n=n+1
01760if n>w1then1840
01770if end:1then1840
01780set1:n
01790input:1,c$
01800if instr(q2$(k),c$)=0then1750
01810q2(k)=n
01820c1=0
01830goto1720
01840gosub3440
01850print" "
01860print"** '";q2$(k);"' is an illegal subject for '";q$(k);"' **"
01870n1=999
01880goto1720
01890if n1=999then150
01900REM ......................... begin search .......................
01910REM  record #1 of the data file contains 'w1'.  This is
01920REM  a pointer to a key search item.  Thus, if the present key
```

```
01930REM  item is a record 'r', the nth key item will be at record 'r+w1*n'.
01940t=0
01950r=0
01960set1:1
01970input:1,q$
01980set2:1
01990ifend:2then2350
02000input:2,c$
02010w1=val(c$)
02020r1=0
02030ifend:2then2350
02040fork=1toi
02050r1=r+q2(k)
02060r1=r1+1
02070set2:r1
02080ifend:2then2350
02090input:2,c$
02100gosub2520
02110nextk
02120ift<>itthen2320
02130t1=t1+1
02140REM 'record' in effect?
02150ifc5<>444then2180
02160print"rec#";r;
02170REM count in effect?
02180ifc2=444then2230
02190ifc1<>444then2320
02200set2:r+2
02210input:2,c$
02220printq$;"=";c$;"    ";
02230REM  detail in effect?
02240ifc1<>444then2320
02250j=r
02260fork=1toi
02270j1=j+q2(k)+1
02280set2:j1
02290input:2,c$
02300printq2$(k);"=";c$;"
02310nextk
02320t=0
02330r=r+w1
02340goto2030
02350print" "
02360gosub3440
02370ift1>1then2410
02380ift1=0then2410
02390print"I have found one form"
02400goto2460
02410ift1<>0then2450
02420ifc7<>444then2450
02430print"I have found no forms"
02440goto2460
02450print"I have found ";t1;"forms."
02460REM 'total' in effect?
02470ifc8<>444then2510
02480fork=1toi
02490print"total ";q2$(k);" = ";q3(k)
02500nextk
02510goto150
02520REM here the actual comparisons are performed.
02530REM The objects of the input sentences (q1$(*)) are compared
02540REM to data according to the verbs in q1(*).
02550p=t
02560n=instr(c$,q1$(k))
02570onq1(k)goto2580,2610,2640,2670
02580ifn=0then2690
02590t=t+1
02600goto2690
02610ifn<>0then2690
02620t=t+1
02630goto2690
02640ifc$<q1$(k)then2690
02650t=t+1
02660goto2690
02670ifc$>q1$(k)then2690
02680t=t+1
02690ifp=tthen2720
02700ifq3(k)=-1then2720
02710q3(k)=val(c$)+q3(k)
02720return
02730REM *************************** create command ***********************
02740n=0
02750p=444
02760scratch:2
02770scratch:1
02780k=1
02790gosub3440
02800print"** when you are finished, type 'done'. **"
02810print" "
02820n=n+1
02830print"item #";n;
02840ifn<>1then2860
02850print"(KEY) ";
02860inputc$
02870ifc$="done"then2910
02880set 1:n
02890print:1,c$
02900goto2820
02910c$=str$(n-1)
02920set2:1
02930print:2,c$
02940goto150
02950REM ************ help command **************
02960print" "
02970gosub3440
02980print"** Commands are as follows: **"
02990print"FIXED COMMANDS:"
03000print"help  code   display  nodisplay  end  exit  fee "
03010print"record  norecord  count  nocount  total  nototal "
03020print"add  create  change  show  word  length  assign"
03030print"VARIABLE COMMANDS:"
03040print"special  find  make  is it possible  suggest "
03050print"is it possible to have   is is possible that"
03060print"match   is there   can there be"
03070goto150
03080stop
03090REM  *********** ADD command **************
03100c$=mid$(c$,4)
03110ifc$<>""then3150
03120gosub3440
03130print"Add what?  (Example: add 4,brown)"
03140goto150
03150set2:1
03160input:2,c1$
03170w1=val(c1$)
03180r=2
03190goto3210
03200r=r+w1
03210set2:r
03220ifend:2then3280
03230input:2,c1$
03240ifinstr(c1$,c$)=0then3200
03250gosub3440
03260print"** '";c1$;"' already exists. **"
03270goto150
03280r1=0
03290ifp<>444then3310
03300r=2
03310r1=r1+1
03320set1:r1
03330ifend:1then150
03340fort=2tow1+1
03350input:1,c$
03360printc$;
03370inputk$
03380ifk$="done"then150
03390set2:r+t-2
03400print:2,k$
03410nextt
03420p=0
03430goto150
03440REM  monkey message
03450print"COMPUTER MESSAGE: ";
03460return
03470stop
03480REM  ***************** CODE COMMAND ****************
03490set2:1
03500gosub3440
03510print"At what record number shall I begin ";
03520inputc$
03530ifc$="done"then150
03540a=val(c$)
03550gosub3440
03560print"At what record # shall I end ";
03570inputc$
03580ifc$="done"then150
03590b=val(c$)
03600fori=atob
03610ifend:2then150
03620set2:i
03630input:2,c$
03640printi;". ";c$
03650nexti
03660goto150
03670REM ******* change command ***********
03680gosub1290
03690c$=mid$(c$,8)
03700ifc$<>""then3750
03710gosub3440
03720print"Change what?";
03730print"(Example: change diet in recipe#1 to nutricious)."
03740goto150
03750n=instr(c$,"in")
03760ifn<>0then3800
03770gosub3440
03780print"illegal 'change' structure: ";
03790goto3710
03800q2$(1)=mid$(c$,1,n-1)
03810c$=mid$(c$,n+2)
03820n=instr(c$,"#")
03830ifn=0then3780
03840q2$(2)=mid$(c$,1,n-1)
03850c$=mid$(c$,n+1)
03860n=instr(c$,"to")
03870ifn=0then3780
03880q2$(3)=mid$(c$,1,n-1)
03890q2$(4)=mid$(c$,n+2)
03900q2$(2)=q2$(3)
03910q2$(3)=q2$(4)
03920r=0
03930r=r+1
03940set1:r
03950ifend:1then4000
03960input:1,c$
03970ifinstr(c$,q2$(1))=0then3930
03980n1=r
03990goto4030
04000gosub3440
04010print"'";q2$(1);"' is an illegal item for this data base."
04020goto150
04030REM have set n=pointer within a data field, find that data field
04040set2:1
04050input:2,c$
04060w1=val(c$)
04070r=2
04080set2:r
04090ifend:2then4140
04100input:2,c$
04110ifinstr(c$,q2$(2))<>0then4170
04120r=r+w1
04130goto4080
04140gosub3440
04150print"I cannot find the entry '";q2$(2);"' in the data."
04160goto150
04170r=r+n1-1
04180set2:r
04190input:2,c$
04200set2:r
04210print:2,q2$(3)
04220gosub3440
04230print"'";c$;"' changed to '";q2$(3);"' in ";q2$(2)
04240goto150
04250REM **** detail command *****
04260c1=444
04270goto150
04280c1=0
04290goto150
04300REM ***** show command *****
04310c4=0
04320c$=mid$(c$,5)
04330ifc$<>""then4370
04340gosub3440
04350print"Show what? (e.g.: show name#J.Smith)."
04360goto150
04370n=instr(c$,"#")
04380ifn=0then4350
04390q2$(1)=mid$(c$,1,n-1)
04400q2$(2)=mid$(c$,n+1)
04410set1:1
04420input:1,c$
04430ifinstr(q2$(1),c$)<>0then4470
04440gosub3440
04450print"'";q2$(1);"' is not a KEY item."
04460goto150
04470set2:1
04480input:2,c$
04490w1=val(c$)
04500r=2-w1
04510r=r+w1
04520set2:r
04530ifend:2then4150
04540input:2,c$
04550ifinstr(c$,q2$(2))=0then4510
04560ifc4=444then4940
04570fori=1tow1
04580set1:i
04590input:1,c$
04600printi;". ";c$;":  ";
04610set2:r+i-1
04620input:2,c$
04630printc$
04640nexti
04650goto150
04660REM ***** count command *****
04670c2=444
04680goto150
04690c2=0
04700goto150
04710REM ******* special command *******
04720c$=mid$(c$,8)
04730ifc$=""then4760
04740chainc$
04750goto150
04760gosub3440
04770print" "
04780printtab(5);"To use your own commands , you may type"
04790print"'special progx', where 'progx' is the name of a program"
04800print"which you wish to call from within this system.  The"
04810print"the second last line of 'progx' must be"
04820print"'chain progq'.  The 'x' in 'progx'"
04830print"may be in the range 1-99, i.e. you may have up"
04840print"one hundred routines of your own."
04850goto150
04860REM ***************** DELETE COMMAND ***********************
04870c$=mid$(c$,4)
04880c4=444
04890ifc$<>""then4370
04900c4=0
```

```
04910gosub3440
04920print"Delete which item.  Delete name=0,Shutin"
04930goto5150
04940set2:r
04950input2,c$
04960gosub3410
04970print"Delete ";c$;
04980inputx$
04990ifmid$(x$,1,1)="n"then150
05000fori=1to+1
05010set2:r+i-1
05020print:2," "
05030nexti
05040gosub3440
05050print"Form deleted for all items ";c$
05060goto0150
05070REM ************ RECORD COMMAND ************
05080c5=444
05090goto0150
05100c5=0
05110goto0150
05120REM ********** Length command ***********
05130gosub3140
05140print"The file contains about ";int(l/2);" items."
05150goto0150
05160REM ****** Sort command ******
05170restore
05180print
05190i=0
05200reada$,b$
05210ifa$="999"then5260
05220i=i+1
05230printa$;" ";tab(i*16);
05240ifi>3then5180
05250goto5200
05260print
05270goto0150
05280REM **** TOTAL COMMAND ****
05290gosub3440
05300print" You may request totals for NUMERIC items only.  For"
05310print"example:  FIND A COST* WHICH IS LESS THAN 50"
05320print"AND A CALORIE-COUNT* OF MORE THAN TEN AND"
05330print"AN AGE OF 15."
05340print" This would present sums of the value of"
05350print"'calorie-count' and 'cost', but not of age."
05360goto0150
05370REM *** A user may have up to ten data bases ***
05380print"Please type the number of your data base";
05390inputn
05400ifn<1then5380
05410ifn>10then5380
05420onnthen5430,5450,5470,5490,5510,5530,5550,5570,5590,5610
05430file:1,"code1.bas$20",:2,"data1.bas$40"
05440goto5620
05450file:1,"code2.bas$20",:2,"data2.bas$40"
05460goto5620
05470file:1,"code3.bas$20",:2,"data3.bas$40"
05480goto5620
05490file:1,"code4.bas$20",:2,"data4.bas$40"
05500goto5620
05510file:1,"code5.bas$20",:2,"data5.bas$40"
05520goto5620
05530file:1,"code6.bas$20",:2,"data6.bas$40"
05540goto5620
05550file:1,"code7$20",:2,"data7.bas$40"
05560goto5620
05570file:1,"code8.bas$20",:2,"data8.bas$40"
05580goto5620
05590file:1,"code9.bas$20",:2,"data9.bas$40"
05600goto5620
05610file:1,"code10.bas$20",:2,"data10.bas$40"
05620goto0130
09033REM ******** Reassign command *******
09202goto0110
09499end
```

Circuit

By Tim Gates

LOGIC.BAS is a simple logic analysis program. Circuits may be entered and tested under changing input conditions. All of the nodes in the circuit are monitored and their states are printed after each cycle.

This program may be used only on logic gates. The following gates are allowable: AND, NAND, OR, NOR, NOT, BUFT, XOR, AND XNOR. Both NOT and BUFR have one input. XOR and XNOR have two inputs each. The other gates, AND, NAND, OR, NOR, may have from 2 to 8 inputs per gate, inclusive.

If more complex functions are desired, construct the function out of allowable gate types.

EXAMPLE CIRCUIT:

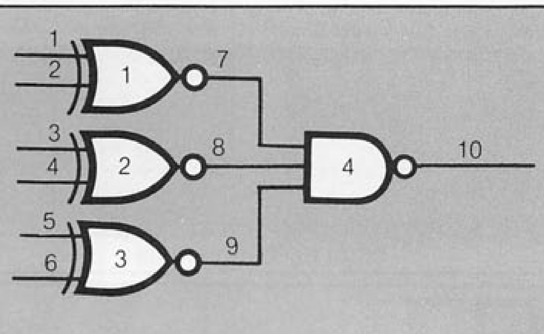


Figure 1. Example Circuit

CIRCUIT GENERATION

To prepare the circuit for analysis:

1. Convert all complex functions to allowed gates.
2. Number all gates in the circuit. Start with 1 and increment for successive gates. The number assigned to a gate does not affect operation. Number the gates in any order, but don't omit any numbers.
3. Number all nodes in the circuit.

The next step is to enter the circuit.

First — Execute BASIC (dated 1-04-78 or later)
Then — EXE LOGIC

When prompted with:

FUNCTION (CIRCUIT,SEQUENCE,ANALYSIS,END)?

Reply:

C

Then hit return.

This executes the circuit entering section of LOGIC. The next question asks for the name of a file in which to store the circuit.

FILE TO STORE CIRCUIT IN? (CR = NO FILE)?

If no file is wanted then hit return. Otherwise type the name of the file the circuit is to be stored in.

Example:

FILE TO STORE CIRCUIT IN? (CR = NO FILE)? FRED

The file 1:FRED.DAT would be used to store the circuit. To the question:

HOW MANY GATES ARE IN THE CIRCUITRY?

type the number of gates in the circuit. For the example circuit the response is '4'. No more than 50 gates may be used.

The gate types will be asked for one at a time. In response to:

GATE # n ?

where n is the number of the gate being entered, type the function of the gate, i.e., AND, NAND, OR . . . etc. Depending on what type of gate is entered, the inputs of the gate will be asked for.

For AND, NAND, OR, NOR the question will be:

INPUT BETWEEN 2 AND 8 NODES INCLUSIVE?

For NOT and BUFR the question will be:

INPUT NODE?

For XOR and XNOR the question will be:

INPUT 2 NODES?

In response to any of the above questions, type the number(s) of the node(s) which are inputs to the gate. When asked:

OUTPUT NODE?

type the number of the output node. This sequence of questions starting with 'GATE # n ?' will repeat until all the gates are entered. For the example, entering would be as follows:

```
GATE # 1 ? XNOR
INPUT 2 NODES? 1,2
OUTPUT NODE? 7
GATE # 2 ? XNOR
INPUT 2 NODES? 3,4
OUTPUT NODE? 8
GATE # 3 ? XNOR
INPUT 2 NODES? 5,6
OUTPUT NODE? 9
GATE # 4 ? NAND
INPUT BETWEEN 2 AND 8 NODES INCLUSIVE? 7,8,9
OUTPUT NODE? 10
```

SEQUENCE GENERATION

After all the gates are entered, the program will return to the question:

FUNCTION (CIRCUIT,SEQUENCE,ANALYSIS, END)?

Type:

S

Then hit return.

This executes the sequence generation section of LOGIC. This section generates the test sequence the circuit is to be put through. The next question asks what file the test sequence is to be stored in:

FILE TO STORE TEST SEQUENCE IN?
(CR = NO FILE)?

Type the name of the file. If 'SEQ' was typed, the file used for storing the test sequence would be '1:SEQ.DAT'. Next, the quantity of nodes in the circuit to be tested is entered.

Analysis

HOW MANY NODES ARE IN THE CIRCUIT?

For the example the answer is '10'.

The next question asks what nodes are to be controlled by the world outside of the circuit. The control nodes are the input nodes to the circuit which are to be set by the test sequence at the beginning of every test pattern.

WHICH NODES OF THE CIRCUIT DO YOU WANT TO CONTROL WHILE THE CIRCUIT IS OPERATING?

Type the number(s) of the node(s) you wish to control. For the example the response is '1,2,3,4,5,6'.

Next, the initial values of the nodes are asked for:

INITIAL VALUE OF NODE x ?

Where x is the number of the node. Type one of the following:

0 for false
1 for true
2 for unknown

Then hit return. Do this until all the nodes have been initialized. For the example the inputs would be:

```
INITIAL VALUE OF NODE 1 ? 0
INITIAL VALUE OF NODE 2 ? 1
INITIAL VALUE OF NODE 3 ? 0
INITIAL VALUE OF NODE 4 ? 1
INITIAL VALUE OF NODE 5 ? 0
INITIAL VALUE OF NODE 6 ? 1
INITIAL VALUE OF NODE 7 ? 2
INITIAL VALUE OF NODE 8 ? 2
INITIAL VALUE OF NODE 9 ? 2
INITIAL VALUE OF NODE 10 ? 2
```

Then in answer to:

NODE x ?

where x is the number of one of the control nodes previously typed, enter a 0, 1 or a 2, depending on the desired value of that particular node during the current step of the test sequence. When all the test steps have been entered, type '-77' in response to any input, and it will terminate the input mode. For the example, this set of inputs is as follows:

```
STEP 2
NODE 1 ? 0
NODE 2 ? 0
NODE 3 ? 0
NODE 4 ? 1
NODE 5 ? 0
NODE 6 ? 1
STEP 3
NODE 1 ? 1
NODE 2 ? 1
   •
   •
   •
NODE 5 ? 1
NODE 6 ? 1
STEP 6
NODE 1 ? -77
```

The program will not return to the question:

FUNCTION (CIRCUIT,SEQUENCE,ANALYSIS,END)?

Type:

A

Then hit return.

This executes the circuit analysis section of the program. In this section the actual circuit analysis takes place. The first question is:

LIST DEVICE?

Type either "*CN' or "*PR' for the console or the printer respectively. This is where the results of the analysis are printed. If there is no printer then type "*CN'. The second question is:

WHAT IS THE MAXIMUM NUMBER OF ITERATIONS UNTIL THE NEXT SET OF TEST CONDITIONS?

This is the number of iterations that will be printed out if the circuit is oscillating. When the circuit stabilizes the next set of conditions is started. For the example type '5'. The third question is:

CIRCUIT FILE?

Type the name of the file which the circuit was stored in. If the circuit was entered during the current execution of LOGIC, hit the return key without typing a file name. For the example type 'FRED'. The last question is:

TEST SEQUENCE FILE?

Type the name of the file containing the test sequence. If the sequence was entered during the current execution of this program, hit return without entering a file name. For the example type 'SEQ'.

The program will now print out the values of the nodes of all the iterations that the circuit is to be put through. Output from the example would be:

```
STEP No. 1
      +          *
0101   01XX   XX
0101   0100   0X
0101   0100   01
0101   0100   01

STEP No. 2
      +          *
0001   0100   01
0001   0110   01
0001   0110   01

STEP No. 3
      +          *
1101   0110   01
1101   0110   01

STEP No. 4
      +          *
0000   0110   01
0000   0111   01
0000   0111   01
```

The sample output shows the values of 10 nodes. Node 1 is at the far left. Node 10 is at the right. The nodes are printed in blocks of four. The next step will be started when the circuit stabilizes or the maximum number of iterations has been reached.

0 = = low or false
1 = = high or true
X = = unknown value

STEP No. 5

	+	*
1100	1111	01
1100	1111	11
1100	1111	10
1100	1111	10

STEP No. 6

+ *

After the test sequence has completed, the program will prompt with:

FUNCTION (CIRCUIT,SEQUENCE,ANALYSIS,END)?

Type:

E

Then hit return. The program will now end.□

PROGRAM LISTING

```
1000 OPTION BASE 0
1020 OPTION LENGTH 100
1040 WIDTH 100
1060 DIM A(70),B(70),D(50,10),M(4,8),S(60,10),T(9,1)
1080 FOR X=0 TO 9
1100   T(X,0)=4^X*3:T(X,1)=4^X
1120 NEXT X
1140 C$="----":X$="01X2":S$="----"
1160 INPUT "FUNCTION (CICRUIT,SEQUENCE,ANALYSIS,END)":A$
1180 IF LEFT$(A$,1)="S" THEN 3060
1200 IF LEFT$(A$,1)="C" THEN 1460
1220 IF LEFT$(A$,1)="A" THEN 4200
1240 IF LEFT$(A$,1)<>"E" THEN 1280
1260 STOP
1280 PRINT "PLEASE USE:"
1300 PRINT " 'A'  FOR  LOGIC ANALYSIS"
1320 PRINT " 'C'  FOR  GENERATING DATA OF YOUR CIRCUIT"
1340 PRINT " 'S'  FOR  GENERATING DATA FOR TEST SEQUENCE"
1360 PRINT " 'E'  FOR  ENDING PROGRAM"
1380 GOTO 1160
1400 '  --------------------
1420 '  = CIRCUIT GENERATOR =
1440 '  --------------------
1460 LINE INPUT "FILE TO STORE CIRCUIT IN? (CR=NO FILE)":C$
1480 IF C$="" THEN 1520
1500 OPEN "O",1,C$ :NG=INT(NG) ' OPEN FILE TO STORE DATA IN
1520 INPUT "HOW MANY GATES ARE IN THE CIRCUIT":NG
1540 IF NG<1 OR NG>50 THEN 1520
1560 IF C$="" THEN 1680
1580 PRINT #1:NG
1600 'D(X,0)   =  GATE TYPE
1620 'D(X,1)   =  NO. OF INPUTS
1640 'D(X,2-9) =  INPUT NODES
1660 'D(X,10)  =  OUTPUT NODE
1680 PRINT "USE ONLY THE FOLLOWING TYPES OF GATES WHEN"
1700 PRINT "PROMPTED WITH 'GATE?' :"
1720 PRINT "AND,NAND.OR,NOR,NOT,BUFR,XOR,XNOR"
1740 PRINT
1760 PRINT "WHEN ASKED FOR INPUT NODE(S), TYPE THE NUMBER(S) OF THE"
1780 PRINT "NODE(S) WHICH ARE INPUTS TO THAT GATE"
1800 PRINT
1820 PRINT "WHEN ASKED FOR THE OUTPUT NODE, TYPE THE NUMBER OF THE"
1840 PRINT "NODE WHICH IS CONNECTED TO THE OUTPUT OF THE GATE"
1860 PRINT
1880 FOR X=1 TO NG
1900   PRINT "GATE #":X::INPUT G$
1920   IF G$= "AND" THEN Y=1:GOTO 2640
1940   IF G$="NAND" THEN Y=-1:GOTO 2640
1960   IF G$= "OR"  THEN Y=2:GOTO 2640
1980   IF G$="NOR"  THEN Y=-2:GOTO 2640
2000   IF G$="NOT"  THEN Y=3:GOTO 2520
2020   IF G$="BUFR" THEN Y=-3:GOTO 2520
2040   IF G$="XOR"  THEN Y=4:GOTO 2160
2060   IF G$="XNOR" THEN Y=-4:GOTO 2160
2080   PRINT "INVALID"
2100   PRINT "USE ONLY THE FOLLOWING TYPES OF GATES"
2120   PRINT "AND,NAND,OR,NOR,NOT,BUFR,XOR,XNOR"
2140   GOTO 1920

2160   'INPUTS FOR XNOR,XOR
2180   FOR J=4 TO 9  ' SET UNUSED LOCATIONS OF D(X,) TO 0
2200     D(X,J)=0
2220   NEXT J
2240   INPUT "INPUT 2 NODES":A,B ' XNOR,XOR  ONLY
2260   IF A<1 OR B<1 OR A>70 OR B>70 THEN 2160
2280   D(X,2)=INT(A):D(X,3)=INT(B)
2300   D(X,1)=2:D(X,0)=Y ' SAVE FUNCTION AND '2' FOR THE No. OF INPUTS
2320   INPUT "OUTPUT NODE":C
2340   IF C<1 THEN 2320
2360   C=INT(C):D(X,10)=C ' SAVE OUTPUT NODE
2380   IF C$="" THEN 2440
2400   '  PRINT DATA INTO FILE (11 NUMBERS)
2420   FOR J=0 TO 10 :PRINT#1,D(X,J) :NEXT J
2440 NEXT X
2460 IF C$="" THEN 1160
2480 CLOSE #1
2500 GOTO 1160
2520 INPUT "INPUT NODE":A 'NOT,BUFR ONLY
2540 IF A<1 OR A>70 THEN 2520
2560 A=INT(A):B=A

2580 GOTO 2280
2600 GOTO 1160
2620 '  MULTIPLE INPUT GATE  (NAND,AND,XNOR,OR  ONLY)
2640 K=1
2660 LINE INPUT "INPUT BETWEEN 2 AND 8 NODES INCLUSIVE":A$
2680 K=K+1
2700 J=VAL(A$):J=INT(J)
2720 IF J<1 OR J>70 THEN 2640
2740 D(X,K)=J
2760 FOR J=1 TO LEN(A$) ' FIND NEXT COMMA IN A$
2780   IF MID$(A$,J,1)="," THEN 2840
2800 NEXT J
2820 GOTO 2880
2840 A$=MID$(A$,J+1)
2860 GOTO 2680
2880 FOR J=K+1 TO 9  ' SET UNUSED LOCATIONS OF D(X, ) TO 0
2900   D(X,J)=0
2920 NEXT J
2940 K=K-1
2960 D(X,1)=K:D(X,0)=Y
2980 GOTO 2320

3000 ' --------------------
3020 ' =  SEQUENCE GEN.    =
3040 ' --------------------
3060 INPUT "HOW MANY NODES ARE IN THE CIRCUIT":NN
3080 IF NN<1 OR NN>70 THEN 3060
3100 ST=1
3120 NN=INT(NN)
3140 LINE INPUT "FILE TO STORE TEST SEQUENCE IN? (CR=NO FILE)":S$
3160 IF S$="" THEN 3200
3180 OPEN "O",2,S$
3200 PRINT "WHICH NODES OF THE CIRCUIT DO YOU WANT TO CONTROL"
3220 PRINT "WHILE THE CIRCUIT IS OPERATING":
3240 LINE INPUT A$
3260 FOR X=1 TO NN:A(X)=0:NEXT X
3280 ' REMOVE NUMBER FROM A$
3300 X=VAL(A$) ' GET A NUMBER

3320 IF X<=0 OR X>NN THEN 3200
3340 A(X)=1
3360 FOR X=1 TO LEN(A$) ' FIND A COMMA IN A$
3380   IF MID$(A$,X,1)="," THEN 3440
3400 NEXT X
3420 GOTO 3480
3440 A$=MID$(A$,X+1) ' SHORTEN A$ TO START WITH NEXT NUMBER IN A$
3460 GOTO 3300
3480 FOR X=0 TO NN-1
3500   PRINT "INITIAL VALUE OF NODE":X+1::INPUT Y
3520   IF Y<>0 AND Y<>1 AND Y<>2 THEN 3500
3540   Z=X-INT(X/7)*7:I=INT(X/7)
3560   S(ST,I)=(S(ST,I) AND (NOT T(Z,0))) OR (T(Z,1)*Y)
3580 NEXT X
3600 IF S$="" THEN 3700
3620 PRINT #2,NN 'PRINT NUMBER OF NODES INTO FILE
3640 FOR X=0 TO INT((NN-1)/7)
3660   PRINT #2,S(ST,X) 'PRINT INITIAL VALUES INTO FILE
3680 NEXT X
3700 ST=2
3720 PRINT "ENTER -77  TO ANY NODE TO STOP SEQUENCE GENERATION."
3740 GOTO 3820
3760 IF S$="" THEN 1160
3780 CLOSE #2
3800 GOTO 1160
3820 PRINT "STEP":ST
3840 FOR X=1 TO NN'INPUT ONLY NODES WHICH ARE INPUTS
3860   IF A(X)=0 THEN 3940
3880   PRINT "NODE":X::INPUT Y
3900   IF Y=-77 THEN 3760
3920   IF Y<0 OR Y>3 OR Y<>INT(Y) THEN 3880
3940   I=INT((X-1)/7):Z=X-I*7-1
3960   IF A(X)=0 THEN 4020
3980   S(ST,I)=(S(ST,I) AND ( NOT T(Z,0))) OR (T(Z,1)*Y)
4000   GOTO 4040
4020   S(ST,I)=S(ST,I) OR T(Z,0)
4040 NEXT X
4060 IF S$="" THEN 4100
4080 FOR X=0 TO INT((NN-1)/7):PRINT #2,S(ST,X):NEXT X
4100 ST=ST+1
4120 GOTO 3820
```

```
4140 '  -----------------
4160 '  = LOGIC ANALYSIS =
4180 '  -----------------
4200 INPUT "LIST DEVICE";P$
4220 OPEN "O",3,P$
4240 C2=2:C3=3:K=1
4260 FOR X=1 TO NN :A(X)=4:B(X)=4 :NEXT X
4280 PRINT "WHAT IS THE MAXIMUM NUMBER OF ITERATIONS UNTIL THE NEXT"
4300 INPUT "SET OF TEST CONDITIONS";T
4320 T=INT(T) MAX 2
4340 LINE INPUT "CIRCUIT FILE";CI$
4360 IF CI$="" AND C$="----" THEN 4340
4380 IF CI$="" THEN 4480
4400 IF CI$<>C$ THEN 4460
4420 CI$=""
4440 GOTO 4480
4460 OPEN "I",1,CI$
4480 LINE INPUT "TEST SEQUENCE FILE";SI$
4500 IF SI$="" AND S$="----" THEN 4480
4520 IF SI$="" THEN 4620
4540 IF SI$<>S$ THEN 4600
4560 SI$=""
4580 GOTO 4620
4600 OPEN "I",2,SI$
4620 ' ARRAY M(4,9) IS ADDRESSED AS A THREE DIMENSIONAL
4640 '    ARRAY.  M(F,A*3+B) = M(F,A,B) = M(4,3,3)
4660 ' THIS ARRAY CONTAINS THE LOGIC TABLE OF THE ALLOWED
4680 '   'GATES (AND,OR,NOT,XOR)
4700 ' THE FOLLOWING GATES (NAND,NOR,BUFR,XNOR) MAKE 2 PASSES
4720 '    THROUGH THE LOGIC ARRAY.  FIRST PASS IS THE ACTUAL
4740 '    LOGIC FUNCTION.  THE SECOND PASS IS TO INVERT THE OUTPUT
4760 '
4780 ' VALUES OF F FOR DIFFERENT LOGIC FUNCTIONS
4800 '    AND=1     NAND=-1
4820 '    OR =2      NOR=-2
4840 '    NOT=3     BUFR=-3
4860 '    XOR=4     XNOR=-4
4880 ' REM ARRAY A LAST NODE VALUES
4900 ' 0=LOW ,1=HIGH ,2=UNKNOWN ,3=DON'T CARE
4920 '  READ LOGIC TABLE
4940 FOR X=1 TO 4
4960   READ M(X,0),M(X,1),M(X,2),M(X,3),M(X,4)
4980   READ M(X,5),M(X,6),M(X,7),M(X,8)
5000 NEXT X
5020 '  READ CIRCUIT
5040 IF CI$="" THEN 5160
5060 INPUT #1,NG
5080 FOR X=1 TO NG
5100    INPUT #1,D(X,0),D(X,1),D(X,2),D(X,3),D(X,4)
5120    INPUT #1,D(X,5),D(X,6),D(X,7),D(X,8),D(X,9),D(X,10)
5140 NEXT X
5160 'READ INITIAL NODE VALUES
5180 IF SI$="" THEN 5260
5200 INPUT #2,NN
5220 Z=INT((NN-1)/7)
5240 FOR X=0 TO Z:INPUT #2,S(K,X):NEXT X
5260 FOR X=0 TO NN-1
5280    I=INT(X/7):Z=X-I*7
5300   B(X+1)=(S(K,I) AND T(Z,0))/T(Z,1)
5320 NEXT X
5340 TM=0
5360 T$="    +    *    *    *"
5380 T$=T$+T$+T$+T$
5400 T$=LEFT$(T$,NN+INT((NN-1)/4))
5420 PRINT #3:"STEP No. 1"
5440 PRINT #3:T$
5460 GOTO 5760
5480 ' MAKE A PASS ON DATA
5500 TM=TM+1
5520 IF TM>=T THEN 5960
5540 FOR X=1 TO NN:A(X)=B(X):NEXT X
5560 FOR X=1 TO NG
5580    F=ABS(D(X,0))
5600    IF D(X,1)>2 THEN 6560
5620    O=M(F,A(D(X,2))*C3+A(D(X,3)))
5640    IF D(X,0)>0 THEN 5700
5660   B(D(X,10))=M(C3,C3*O)
5680   GOTO 5720
5700   B(D(X,10))=O
5720 NEXT X
5740 L$=""
5760 FOR X=1 TO NN ' PRINT VALUES OF NODES
5780    L$=L$+MID$(X$,B(X)+1,1)
5800    IF X<> INT(X/4)*4 THEN 5840
5820    L$=L$+" "
5840 NEXT X
5860 PRINT #3:L$  'PRINT LINE
5880 FOR X=1 TO NN
5900    IF A(X)<>B(X) THEN 5480
5920 NEXT X
5940 ' STEADY STATE
5960 ' READ NEW NODE VALUES
5980 IF EOF(2) THEN 1160
6000 K=K+1:TM=0
6020 PRINT #3
6040 PRINT #3:"STEP No.":K
6060 PRINT #3:T$
6080 IF SI$="" THEN 6160
6100 Z=INT((NN-1)/7)
6120 FOR X=0 TO Z:INPUT #2,S(K,X):NEXT X
6140 GOTO 6180
6160 IF K>=ST THEN 1160
6180 FOR X=0 TO NN-1
6200    I=INT(X/7):Z=X-I*7
6220   Y=(S(K,I) AND T(Z,0))/T(Z,1)
6240    IF Y=3 THEN 6280
6260   B(X+1)=Y
6280   A(X+1)=4
6300 NEXT X:GOTO 5740
6320 ' DATA FOR LOGIC ARRAY
6340 '   0=FALSE
6360 '   1=TRUE
6380 '   2=UNKNOWN
6400 '              AND
6420 DATA  0,0,0,0,1,2,0,2,2
6440 '              OR
6460 DATA  0,1,2,1,1,1,2,1,2
6480 '              NOT
6500 DATA  1,1,1,0,0,0,2,2,2
6520 '              XOR
6540 DATA  0,1,2,1,0,2,2,2,2
6560 J=D(X,1)
6580 ' 3 TO 8 INPUT GATE (AND,NAND,OR,NOR ONLY)
6600 L=3
6620 A=A(D(X,C2)):B=A(D(X,C3))
6640 O=M(F,A*C3+B)
6660 J=J-1:IF J=1 THEN 5640
6680 L=L+1:O=M(F,O*C3+A(D(X,L)))
6700 GOTO 6660
6720 END
```


CIRCLE INQUIRY NO. 64

Personal Computer

Many sources can cause interference on computers. Lightning, heavy machinery and power outages often create damaging power line surges which may damage expensive microprocessors and peripherals. Hash is created by hundreds of sources including spherics, tools, appliances, microprocessors, business machines and defective wiring. It also causes those "glitches", so frustrating to personal computerists. This article discusses basic causes of hash and surges, effects these phenomena have on microcomputer performance, and suggests cures which can be incorporated to reduce undesirable effects.

HASH

Arcing in various types of electrical equipment is responsible for a lot of power line interference, and tools, motors, appliances, and other small electrical devices are notorious offenders. Microprocessors, peripherals and business machines are often guilty of hash generation. Another source of hash is power lines which sometimes function as an antenna, picking up various signals and creating unwelcome interference. Broadcast stations, 2-way radio and CB sets can also be the source of this type interference.

Another common source of hash can be found right in the computerist's home or office. Over a period of time light sockets, wall sockets, line-cord plugs or wire connections often can become loose, defective or corroded and these common and simple sources are overlooked.

Often this hash is well camouflaged. It disappears when a suspected source is shut off. Hours or days later, various schemes have not cured the hash, and it is reluctantly accepted as unavoidable. Actually, shutting off the noisy device stopped an electrical current flow through the noisy connection or socket, and all traces of hash have stopped!

When investigating hash or interference problems don't overlook:

1. SCR or Triac controlled lights, motors and power supplies
2. Fluorescent lamps
3. Teletype
4. Floppy drives
5. Welders
6. Diathermy
7. Spherics
8. Noisy electrical sockets and connections
9. Nearby internal combustion engines

EFFECTS OF HASH

As relates to personal computers, hash takes on two dimensions. First, externally created hash interferes with smooth microprocessor operation. This is the case where "glitches" foul up the Space Wars Game, disrupt a program previously proven to be faultless or create erroneous printout. Garbled communication or erratic processor/peripheral behavior are often the result of externally created hash. Figure 1 presents a printout with hash induced glitches.

A second dimension of the hash problem exists when the microprocessor and/or peripherals create hash which interfere with some external device. Hearing aids, heart pacers, CB radios, audio equipment and TV have all had interference problems traceable to microprocessors.

MAKE PAYMENT TO	AMOUNT DUE
ELECTRONIC SPECIALISTS, INC 171 S MAIN ST NATICK, MASS 01760	$169.75
MAKE PQYMENT TO	AMOUNT TUE
ULECTrONIC SpECIALISTS, INC 171 S MAIN ST NATICK, MASS 01wv0	4169.75

Figure 1. Garbled printout due to line hash.

POWER LINE HASH CURES

Hash filters are often installed at the microprocessor suffering from hash interference, and often this completely eliminates interference problems. However, elimination of hash at the source is a more desirable approach.

Lamp sockets and electrical outlets should be investigated. Any defective wiring or components should be repaired or replaced.

Noisy tools, appliances or equipment should be well filtered. Electronic Specialists manufactures a convenient line cord hash filter which can be utilized at either the microprocessor or the hash producing equipment. A wire-in filter for installation directly within the microprocessor, power supply or peripheral as shown in Photo 1. Models are also available which incorporate both a Line Hash Filter and Surge Suppressor.

Another way to handle hash is to make sure all equipment covers and shields supplied by the manufacturer are securely fastened in place. It sometimes becomes necessary to fabricate and install a suitable shield. This can be effective both at keeping external interference out of the microcomputer system and keeping internally generated interference from escaping to create interference elsewhere in the neighborhood.

AC power should be brought into the microprocessor equipment through a 3-wire cable. The third wire (green) should be connected to the equipment chassis. Check to be certain house or office electrical wiring returns the third wire to a good ground (earth).

Often connecting microprocessor and peripheral chassis or cabinets to ground will reduce interference from outside sources and prevent internally generated hash from escaping to cause trouble. When installing a ground system, be careful to avoid ground loops, as these may induce glitches caused by system hum. Generally, it is advisable to start with one piece of equipment tied to a good ground, then proceed with a second piece tied to the same ground, and so on. Check system operation after each piece is connected to ground.

AC POWER LINE SURGES

Large, damaging surges and transients on the AC power line are often caused by lightning. Power line voltages may momentarily increase to 5 or 10 times the normal AC line voltage. Estimates place the electrical potential of lightning at several million volts and the arc or "bolt" current up to one million amperes. Instantaneous power of over one trillion watts is produced!

Protection

By F. J. Stifter
President, Electronic Specialists, Inc.

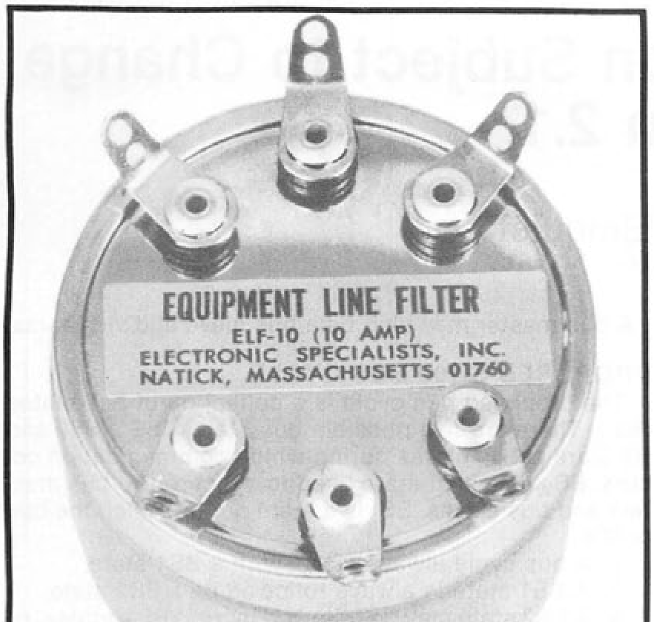


PHOTO 1 Wire-in Hash Filter and Surge Suppressor

That's over 100 million horsepower! Considering that huge power and current flow, it's easy to see why a direct hit is not required to produce a voltage surge many times the normal line voltage.

Significant surges are also created when heavy machinery or switch gear is operated. These surges are not as impressive as lightning but can cause considerable equipment damage. Line surges may reach 2 or 3 times the normally applied line voltage. Power outages are becoming more common, however, and surges frequently accompany loss or restoration of power during an outage.

EFFECTS OF POWER LINE SURGES

Most often, power line surges occur in the Differential Mode. In this mode, short surges of extremely high voltage are developed between the AC lines. These surges can be caused either by lightning or heavy machinery, and behave much like a very high line voltage.

When a Differential Mode power line surge occurs, extremely high voltage is applied to motors, power supplies, lamps or anything that is connected to the AC line. Motor windings may arc over and cause permanent damage. Power transformers develop high secondary voltages, and the power transformer may short, rectifiers may be damaged, and voltage regulators may be destroyed. A "domino effect" could wipe out large sections of the microprocessor.

In some cases the induced voltage may be high enough to arc across "OFF" switch contacts. The same disastrous results may occur even though the equipment had been turned "OFF".

A second type of line surge which often occurs is the Common Mode surge. In Common Mode voltage surges, both AC lines are brought to a very high voltage. This situation is usually caused only by lightning.

When applied to equipment, the Common Mode high voltage may cause arcing between conductors and ground. Insulation of power transformers may be punctured, rendering the unit worthless. Motor windings may arc to the frame, destroying the device. Cable insulation may be punctured; switches and controls may be damaged.

Heavy machinery surges or power outage surges are usually less damaging than lightning induced surges. Although less severe, these surges may still cause costly damage.

Damage caused by power line surges often has a ripple effect resulting in the immediate and permanent damage of a number of components. In addition, a large number of other components either develop reduced performance characteristics or have shortened life-spans. After the initial repair, components will continue to fail, requiring repeated downtime for service, and unaccountable "glitches" become more numerous.

In less severe cases, only the secondary effect occurs. No immediate permanent damage is apparent. Components begin to fail at a more rapid rate than normal and "glitches" suddenly begin appearing in a system which had previously been operating flawlessly.

AC LINE SURGE CURES

Unfortunately no complete cure exists for power line surges. However, modern devices provide the means to obtain a large measure of protection against most surge damage. Zeners, thyrectors, gas-tubes, varistors, spark gaps or other forms of lightning arrestors can be installed at microprocessors and peripherals to prevent line surge damage.

Ideally, these devices should be placed across the AC line and between each AC line and ground. When selecting such devices, care must be exercised to choose components with adequate surge handling capabilities; peak voltage ratings must be suitable for the line being protected; proper bi-polar operating characteristics must be selected; and operating speed must be suitable to provide adequate protection.

Limited protection can be obtained by placing high power zeners across power supply secondaries. This will offer some protection to rectifiers, voltage regulators and electronics. The power transformer, cabling, switches and controls are unprotected.

High power zeners connected at the rectifier output will provide some protection for the voltage regulators and electronics. However, rectifiers, power transformers, cabling, switches and controls will be vulnerable.

Several surge suppressors are available from Electronic Specialists, Inc. A convenient AC line cord model (Photo 1) can be used directly at the microprocessor or peripheral to be protected. Protection can be added without altering the equipment. A wire-in model (Photo 2) is designed to be installed internally to microprocessors, power supplies and peripherals. Both types provide differential (line-to-line) and common-mode (both lines to earth) protection.

Many personal computer owners add surge protection to their equipment as low-cost insurance. Valuable microprocessors and peripherals can be protected against the ravages of power line surges for a low, one-time investment in surge suppressors. Models of these units are available which combine surge suppression and power line hash filtering in one convenient package.

SUMMARY

Causes of AC power line surges and hash have been discussed and various effects upon equipment operation were presented. As an aid to control the ill effects of both line surges and hash, several cures were also offered. □

A Designer's Notes On The S-100 Bus Standard Proposal

Preliminary Specification Subject to Change Version 2.1

By Kells A. Elmquist
Ithaca Audio

There has been a great deal of emphasis placed on industry standards. The S-100 bus, as it is known today is a defacto standard without any real definition. Consequently, the time has come to define the bus in order that all so-called S-100 bus devices are compatible.

INTERFACE AGE Magazine is making this proposed standard available to the readers, and further suggests that it be copied and reprinted as often as necessary. The author would like users and manufacturers to comment on the standard. He can be reached by writing to Ithaca Audio, P.O. Box 91, Ithaca, NY 14850, or calling (607) 273-3271. —*Editor*

PREFACE

This paper is NOT the S-100 Bus Standard. It is intended only as an explanation of our proposals to the Standards Committee, and as an explanation of other proposals being considered by the committee. Our desire is to generate input for the Standards Committee, and therefore all response and criticism is warmly welcomed and will be considered in detail.

Many thanks to the design staff at Ithaca Audio for their exhaustive technical assistance, especially from Steve Edelman for his editing and creative approach to all problems. Thanks also to Community Animation, Inc. for the preparation of the graphic material.

INTRODUCTION

The S-100 bus is a collection of signal buses defined relative to a current bus master. The signal buses are: Address bus, Data In/Data Out bus, Status bus, Control Output bus, Command Input bus, Auxiliary Control bus, Utility bus, Auxiliary DMA bus, Auxiliary Vectored Interrupt bus, and a Front Panel connector.

This division of the signal lines represents a change from the defacto division which was derived from signal groupings on the 8080 processor chip, and has been done both to facilitate the inclusion of lines not generated by 8080 processors, and to generalize the interrelationships of signal groups.

BUS SIGNAL TYPES

There are three types of signals on the S-100 bus:

(M) stands for Bus Master. All signals designated (M) *must* be generated by the current bus master.

(S) stands for bus Slave. A bus slave is only required to generate that subset of type S signals required to communicate with bus masters.

(B) stands for Bus. Signals not of type (M) or of type (S) are by default type (B). Type (B) signals include:

a) Utility signals
b) Signals generated by some but not all bus masters.

A bus master may also be a bus slave and vice versa.

PROPOSED S-100 BUS CYCLE

The proposed bus cycle is a collection of bus states (BS). There are five possible bus states: BS1, BS2 and BS 3 are active states during which communication occurs. BS_W is a wait state used to synchronize bus masters and bus slaves, BS_i is a state during which the bus is idle.

1. A bus cycle always starts with a BS1 state.
2. A BS1 state is always followed by a BS2 state.
3. A BS2 state may be followed by zero BS_W states, or by an arbitrarily large number of BS_W states.
4. A BS3 state follows the last BS_W state, or a BS2 state if there are no BS_W states.
5. A BS3 state is followed by zero to an arbitrarily large number of BS_i states. A BS3 or BS_i terminates a bus cycle.

BUS STATE COMMENTS

BS1 is the bus state during which the address lines and status lines are changing to their values for the current cycle. PSYNC is true beginning with the second half of the state, indicating the beginning of a new bus cycle.

BS2 is the bus state during which address, status, and ready signals become stable. The earliest moment during which address and status information may be sampled varies from processor to processor. To accommodate these variations without penalty to one processor or another, the 8080 clock signal Ø1 has been redefined as Status Strobe, STSTB. This signal, when logically anded with PSYNC, indicates the earliest moment that status may be sampled from the status bus and that address may be sampled from the address bus. Such a redefinition requires no modification to existing 8080 processor boards that use the status strobe output from an 8224 clock generator chip to clock the status latch. This strobe precedes the Ø1 signal by approximately 50 ns., allowing some bus settling time.

Interfacing other processors to the S-100 bus is greatly simplified by the inclusion of the STSTB signal. As it may fall anywhere within the PSYNC interval, it is asynchronous to Ø and can be used to tailor the bus timing to any processor, without sacrificing the simplicity and clarity of the synchronous basic cycle.

8080 processors gate status information onto the data bus during the PSYNC interval. Other current processors do not do this, and it seems future processors (i.e. 16 bit chips) will multiplex address and data rather than status and data. Therefore, it is not required of bus masters to gate status information onto the data bus during the PSYNC interval, nor is it acceptable procedure for a bus slave to examine the data bus for status information.

The second half of a BS2 state begins the data transfer, which is completed during a BS3. If BS_W states are present, all address lines, status lines, and data lines must remain stable during the BS_W states.

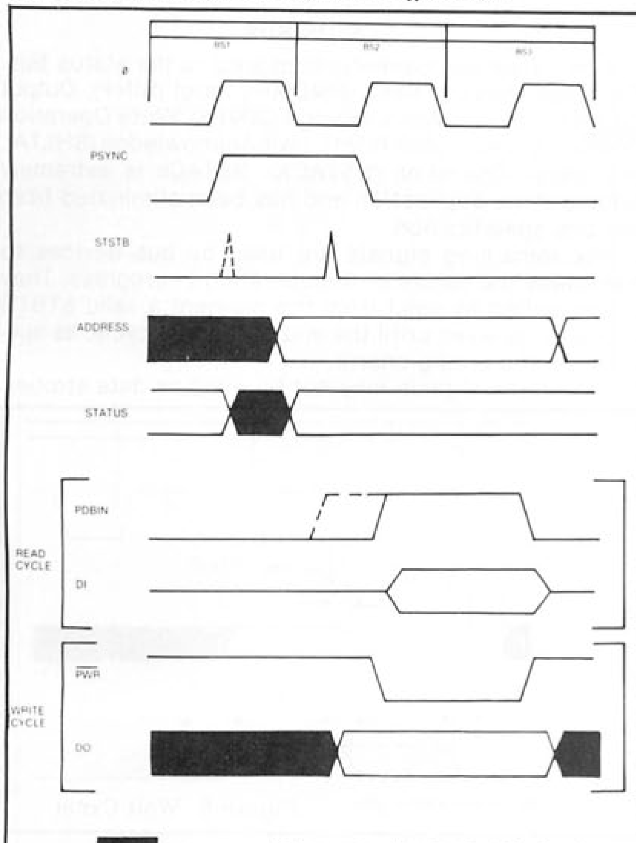


Figure 1. Basic S-100 Cycle

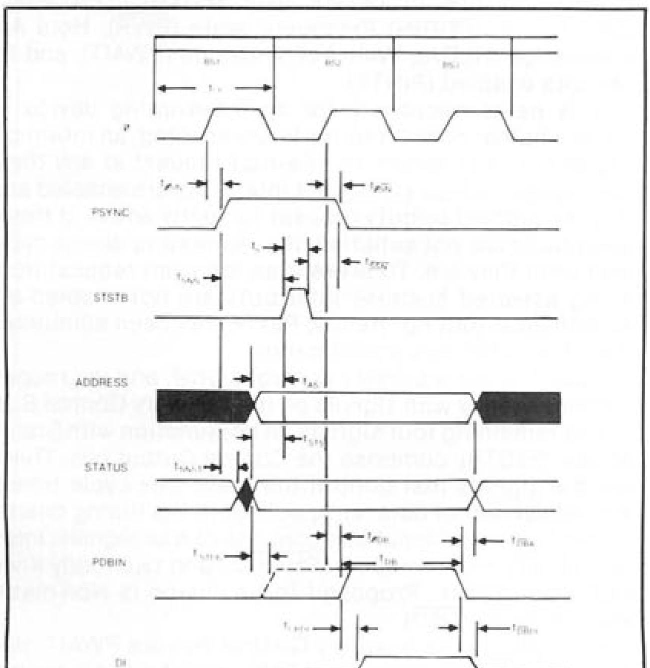


Figure 2. Read Cycle

READ CYCLE

PDBIN is the processor control signal that gates requested information onto the DI bus. With 8080 processors, PDBIN must be delayed until the completion of the PSYNC interval, in order to avoid conflict with status information on the data bus. PDBIN is therefore specified as occurring not later than the 8080 specification, but may occur earlier, shortly after status is valid on the status bus.

The processor latches the data presented during a read cycle before it releases the DBIN signal. Hence, by the end of the DBIN interval, the processor has acquired its data and no hold time is necessary before allowing the address bus to change. This presents a minor problem for devices that wish to "snatch" data off the data bus during a read cycle. Such devices may not use the trailing edge of PDBIN to latch data, since without margin for gate delays, the address bus may already be changing.

The width of the PDBIN interval is specified as not less than one clock period.

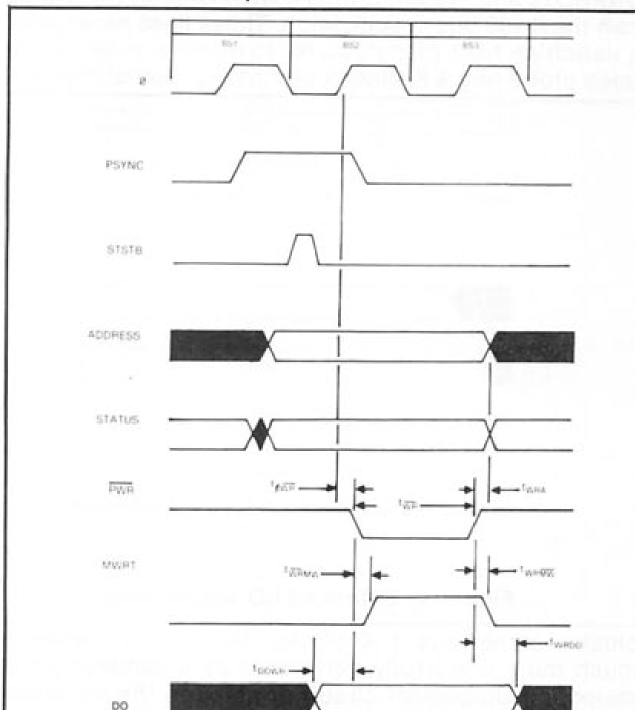


Figure 3. Write Cycle

WRITE CYCLE

The write strobe $\overline{PWR}$ is specified as starting not earlier than the second half of the BS2 state. This allows a half cycle (approximately) from the time that address and status information is stable. The write strobe may, however, start later, allowing extra settling time.

Note that data on the data out bus is specified as valid 100 ns. before the leading edge of the write strobe, as well as 100 ns. after the trailing edge. Therefore, data may be latched on either edge of the write strobe. Address and status information must also be held stable for a specified period after the trailing edge of the write strobe.

The width of the $\overline{PWR}$ strobe is specified as not less than one clock period.

SIGNAL BUS COMMENTS

ADDRESS BUS

The address bus presently consists of sixteen lines, A0-A15. These lines represent a primary address space of 64 K bytes. Various proposals exist for extending the address space, bank select, etc., but the most practical of these is to assign extra address signals. The IEEE committee has proposed the addition of 8 address lines, giving an address space of 16 megabytes.

Input and output devices have been generally addressed by duplicating the 8 bit device address on the high order (A8-A15) and on the low order (A0-A7) address bytes. This is done internally with 8080 processors and must be done with external support circuits for other bus masters. Some processors, the Z-80 and some of the newer 16 bit chips, offer extended I/O addressing,

using the high order address byte to extend the device address. Use of this mode will cause errors with devices designed to examine the high order address byte for the device address. There are two solutions to this problem, neither of which is very attractive. The first is that duplication of the device address be eliminated. This would obsolete many current board designs. The second solution specifies that the extended device address be placed on the extended address byte, A16-A23. Though this requires more support circuitry on the processor board, it seems the least offensive solution.

The memory protect lines, Protect ($\overline{PROT}$), Unprotect ($\overline{UNPROT}$), and Protect Status ($\overline{PS}$) have been eliminated from the S-100 bus specification. These lines have varied in definition from manufacturer to manufacturer, sometimes protecting 4 K blocks of memory, sometimes 8 K blocks, sometimes 1 K blocks. Memory protection is much more powerfully performed by a comprehensive memory management circuit or chip, on the processor board, which combines the functions of memory protection with extended addressing, dynamic relocation, and other memory management functions. Development of such memory management devices is nearing completion at a number of large manufacturers.

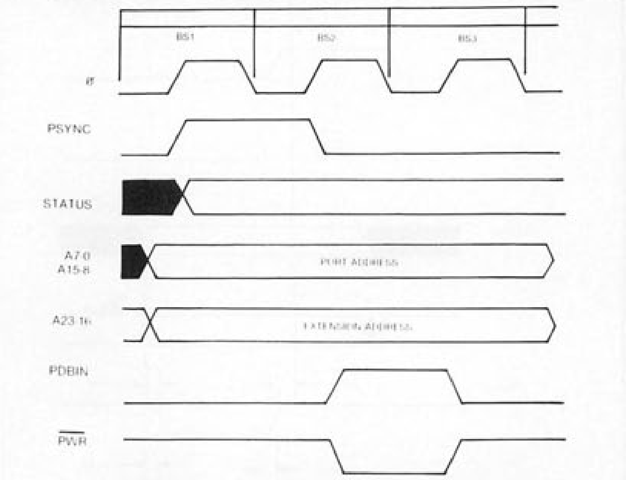


Figure 4. Extended I/O Addressing

DATA IN/DATA OUT BUS

Data input and data output are always specified relative to a bus master. 8 bit data which is transmitted by a bus master is always on the data out bus. 8 bit data which is received by a bus master is always on the data in bus.

A proposal has been included for the specification of 16 bit read/write operations on the S-100 bus. The data in bus and the data out bus are ganged bi-directional buses during 16 bit operations, the low order byte on the DO bus and the high order byte on the DI bus. This technique requires the addition of two lines to the bus definition, Sixteen Request ($\overline{SXTRQ}$), and Sixteen Acknowledge ($\overline{SXTN}$), and memory boards of a unique design, if 16 bit parallel operations are performed. 8 bit operations are performed in the conventional manner, so that both 8 bit and 16 bit masters may co-exist in the same bus. It should be noted at this point that conventional memory boards may still be used in such a system for 8 bit parallel operations or 16 bit byte serial operations. We feel that the ability for both 8 and 16 bit masters to co-exist in the S-100 bus is crucial to the advancement of system architectures. In multi-microprocessor systems and in the case of "smart" peripheral processors we may freely intermix 8 and 16 bit devices, optimizing the bus interface to each specific function.

Sixteen Request is asserted by a 16 bit master with its address, while Sixteen Acknowledge is a response from the addressed slave indicating that a 16 bit parallel operation is possible. If Sixteen Acknowledge is not activated by the addressed slave, the processor may perform the operation in byte serial mode or trap to an error routine.

STATUS BUS

Eight lines are currently assigned to the status bus. They are: Memory Read (SMEMR), Input (SINP), Output ($\overline{SOUT}$), Interrupt Acknowledge (SINTA), Write Operation ($\overline{SWO}$), Op-code fetch (SM1), Halt Acknowledge (SHLTA), and Stack Operation (SSTACK). SSTACk is extremely limited in its application and has been eliminated from the bus specification.

The remaining signals are used by bus devices to determine the nature of the operation in progress. They are specified as valid from the moment a valid STSTB signal is received until the end of the bus cycle, as specified in the timing charts.

The status signals may not be used as data strobes.

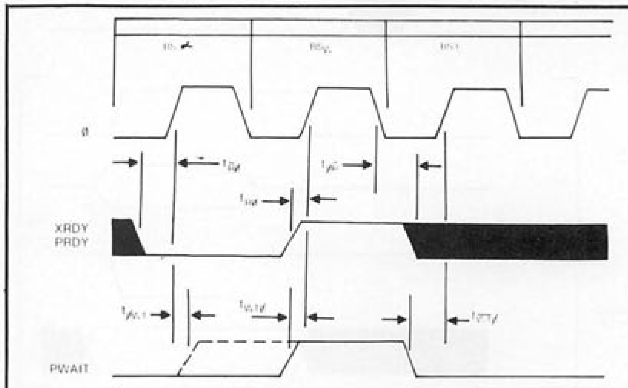


Figure 5. Wait Cycle

COMMAND/CONTROL BUS

The six signals currently assigned to the command/control bus are: Processor Sync (PSYNC), Processor Data Bus In (PDBIN) Processor write ($\overline{PWR}$), Hold Acknowledge (HLDA), Wait Acknowledge (PWAIT), and Interrupts enabled (PINTE).

It is never necessary for an interrupting device to know whether or not interrupts are enabled; an interrupting device may assert an interrupt request at any time, the request will be serviced if interrupts are enabled and it is the highest priority request currently active. If these conditions are not satisfied, the requesting device must wait until they are. To prevent an interrupt request from being asserted because interrupts are not enabled accomplishes nothing. Hence, PINTE has been eliminated from the S-100 bus specification.

PWAIT is not a primary control signal, and is grouped for consistency with signals on the Auxiliary Control Bus.

The remaining four signals, in conjunction with Status Strobe (STSTB), comprise the Control Output bus. These are the signals that control the basic bus cycle timing and movement of data, as specified in the timing charts.

The Command Input bus consists of four signals: Interrupt ($\overline{PINT}$), Hold Request ($\overline{PHOLD}$), and two ready lines XRDY and PRDY. Proposed for inclusion is Non-maskable Interrupt ($\overline{NMI}$).

Included on the Auxiliary Control Bus are PWAIT, Bus Available ($\overline{BA}$) for cycle steal DMA, and the 16 bit control lines $\overline{SXTRQ}$ and $\overline{SXTN}$. The Refresh signal, useful in interfacing dynamic memory, may be included in the S-100 bus specification. If included it would be grouped with the auxiliary control signals.

AUXILIARY DMA AND VECTORED INTERRUPT BUSES

A number of questions have arisen concerning priorities of interrupt and direct memory access devices. The vectored interrupt lines that have been dedicated on the

S-100 bus were designed to operate with first generation interrupt controller chips, which would accept the highest priority request and place an appropriate RST instruction on the data in bus during the interrupt acknowledge cycle. Though a few controller boards have been designed, these vectored interrupt lines have never been widely used and even the active polarity remains in question.

The more intelligent peripheral devices become, the less acceptable is such a scheme. We now have "smart" peripheral chips that generate a variety of interrupt vectors depending on the internal condition that requested the interrupt, but two problems exist that prevent their implementation on the S-100 bus.

First, if an interrupt controller chip is used to prioritize the interrupt requests, a conflict will exist between the vector asserted by the interrupting device and the vector asserted by the interrupt controller. Second, if an interrupt controller chip is not used to prioritize interrupts, no way exists to determine which of the vectored interrupt requests is being acknowledged by the INTA signal.

A similar situation occurs with direct memory access devices. At present it is not possible to resolve conflicts among simultaneous requests for bus control, thus limiting the number of DMA devices allowed on the S-100 bus to one.

To resolve these conflicts a number of lines have been proposed for inclusion in the S-100 bus specification. Eight lines are proposed for inclusion as vectored DMA request lines (DMARQ0-DMARQ7), allowing eight temporary bus masters to co-exist in the S-100 bus; and three lines have been proposed as encoded response lines common to both interrupt and DMA acknowledge cycles. These encoded response lines may be decoded in conjunction with INTA by an interrupting device and compared with its own request vector to determine if it has been acknowledged. A DMA device decodes the response lines in conjunction with HLDA to make a similar determination.

It should be noted that the addition of these lines to the bus does not obsolete the use of interrupt controller chips and the devices designed around them, as the response lines may be ignored in systems using such devices. It is then the responsibility of the interrupt controller to assert the interrupt vector to the central processor.

UTILITY BUS

The following signals are proposed for inclusion on the utility bus:

Power lines: +8 volts, +16 volts, −16 volts.

Ø, system clock.

CLOC, 2 mhz square wave with no specified relationship to Ø.

$\overline{\text{PHANTOM}}$, for overlayed memory.

$\overline{\text{POWER DOWN}}$, indicates an impending power failure.

STANDBY POWER, 5 volt power signal that is independent of the system power supply, useful for powering time of day/calendar chips and power-down mode memories.

$\overline{\text{POC}}$, power on clear signal (*not* debounced reset).

$\overline{\text{RESET}}$, this is the system reset signal before debouncing by the processor board.

$\overline{\text{EXTCLR}}$, this is a clear signal to bus slave devices from the front panel. It is not debounced.

FRONT PANEL CONNECTOR

The connector between the front panel and the CPU card has traditionally been a 16 pin DIP connector, bussing the bi-directional data bus (D0-D7) directly from the CPU chip to the front panel. Three bus lines, RUN, SINGLE STEP, and SENSE SWITCH DISABLE, have been used to control the imposition of instructions and data from the front panel onto the bi-directional data bus. To all CPU cards, these three lines mean exactly the same thing: They gate off the data input drivers so that data from the front panel connector may be read by the CPU. Therefore we should be able to combine these three lines into one and, since they are concerned only with the front panel connector, it is not necessary for this signal ($\overline{\text{DATA IN DISABLE}}$) to appear on the S-100 bus proper, but rather should be specified as part of the front panel connector. Further, we must consider specification of the front panel connector for the new 16 bit machines and, last, we must consider the interconnection problems between the CPU and the minimal front panels in so-called "front panel-less" machines.

We propose a solution to these problems by specifying the front panel connector as a 20 pin DIP plug, with data 0 to data 7 on pins 1-8, data 8 to data 15 on pins 11-18. This organization allows all current 16 pin front panel connectors to be used without modification on 8 bit machines, with the exception of Data In Disable, which is still a bus signal. The quantity of machines that use all three of the signals, run, single step, and sense switch disable, to perform different cases of the same function precludes an immediate change over, but we may accomplish the change by phases, first combining the three into one bus signal, then moving it to the front panel connector.

One function which a number of CPU cards offer is a "power-on-jump". There is at present no signal on the bus that may function as a "jump enable" or "jump" signal, and most CPU cards permanently enable or disable the jump. Pin 9 of the twenty pin front panel connector may be specified as "JUMP", to regain front panel control of this function.

Interconnection between the CPU and a minimal front panel requires two signals besides "JUMP". They are RESET and GROUND. These may be specified on pin 10 and 20 respectively. Specifying these signals as part of the front panel connector allows mass termination cables to be designed for all configurations of front panels.

The $\overline{\text{DATA IN DISABLE}}$ signal is specified on pin 19 of the front panel connector, and, for the time being, is duplicated on the bus, replacing the $\overline{\text{SSWDSB}}$ line. RUN and SS have been eliminated from the bus specification. Boards which need to know whether the CPU is in the "run" mode should examine the ready line driven by the front panel, usually XRDY.

FRONT PANEL CONNECTOR

1) DO	11) D8
2) D1	12) D9
3) D2	13) D10
4) D3	14) D11
5) D4	15) D12
6) D5	16) D13
7) D6	17) D14
8) D7	18) D15
9) $\overline{\text{JUMP}}$	19) $\overline{\text{DATA IN DISABLE}}$
10) $\overline{\text{RESET}}$	20) GROUND

DMA CYCLE DESCRIPTION

A DMA cycle is a special case of a temporary bus master taking control of the bus from the permanent bus master to execute a read or write cycle. The difference between a temporary bus master and a permanent bus master is that,

1. a temporary master is not subject to interrupts, and
2. a temporary master is not subject to a hold operation (no nested DMA).

A temporary master is required to generate *all* type M (bus Master) output signals.

It should be noted that the system clock, Ø, is *never*

exchanged during a bus exchange between permanent and temporary masters. The clock is a utility signal and all masters must create their bus cycle timing from it.

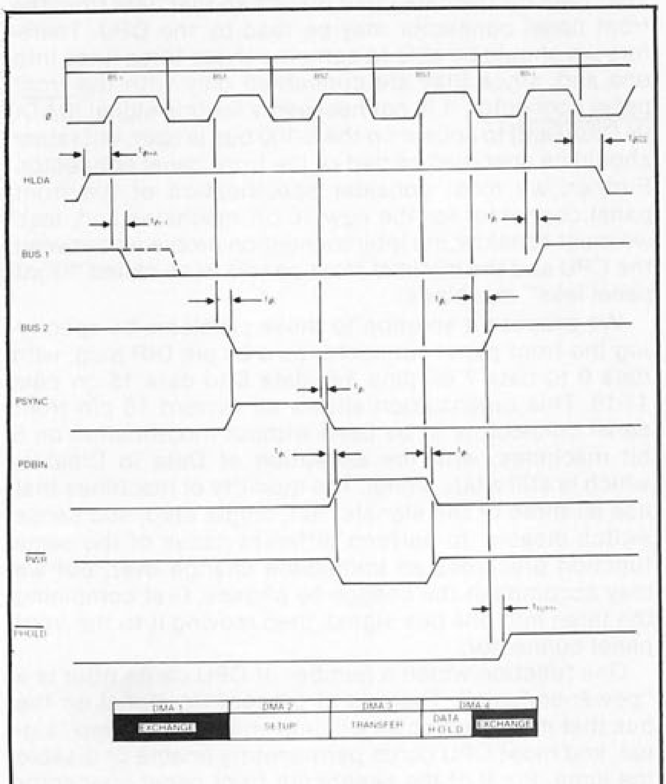


Figure 6. DMA Cycle

THE DMA CYCLE

The DMA cycle has been divided into four equal time blocks (DMA1-DMA4) beginning with the rising edge of Ø following the assertion of a true Hold Acknowledge signal, HLDA. Each time block lasts one clock period. The DMA 1 period is that during which the bus exchange takes place. The signal BUS1 has replaced both the adress disable (ADDDSB) and the data out disable (DODSB) signals, since they are asserted together. The BUS 1 signal is asserted within the area shown following a HLDA = true. BUS 1 turns off the data output drivers and the address drivers of the permanent master and turns on the Control output drivers of the DMA device. *Both* the DMA device and the permanent master are required to drive the Control output lines during the bus exchange operation, thus assuring a smooth transition of the positive polarity control signals. During the times the control lines are driven by both masters, they are required to have the following levels:

PSYNC = 0
PWAIT = 0
PDBIN = 0
PHLDA = 1
PWR = 1

The next rising edge of Ø begins the DMA 2 period, and the BUS 2 signal is asserted. The BUS 2 signal is a combination of the Status Disable (STATDSB) and the Command/Control Disable (C/CDSB). The assertion of the BUS 2 signal turns off the Status drivers and the Control Output drivers of the permanent master, and turns on the Data Output, Address, and Status drivers of the temporary master. The situation at this point is equialent to the second half of a BS 1 state, with the temporary master in complete control of the bus. PSYNC goes active and a normal read or write cycle follows during DMA 2 and DMA 3.

The first half of the DMA 4 period is hold time for data and address lines, and during the second half of cycle the inversion of the bus exchange sequence occurs. BUS 2 is released on the falling edge of Ø, turning off the Data Ouput, Address, and Status drivers of the temporary master and turning on the Control Output drivers of the permanent master. Both devices drive the Control Output lines until the next rising edge of Ø, when BUS 1 is released. The PHOLD signal is released with the BUS 2 signal to avoid adding an extra BS_i state to the DMA cycle.

A DMA operation may be performed where more than one read/write operation is executed. All timing remains the same, the exchange operation "brackets" any number of bus cycles, each of which is subject to the basic bus cycle timing.

It should be noted that in multiple priority DMA systems, the device that arbitrates among the requesting devices may be responsible for the timing of all bus exchange operations if desired. This would relieve the system of considerable duplication of circuitry.

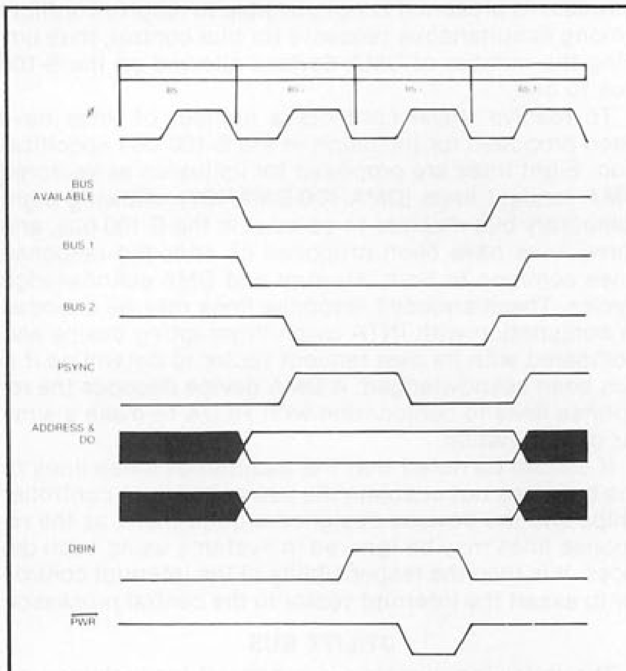


Figure 7. Transparent Data

TRANSPARENT DMA

A transparent DMA cycle is defined as a cycle during which the DMA device takes control of the bus without placing the processor in a HOLD condition. It requires a bus signal indicating that the bus will be available for at least the next two clock periods. The cycle is essentially a condensed version of the standard DMA cycle.

The BUS AVAILABLE signal sets the BUS 1 signal and the BUS 2 follows on the next rising edge of Ø. The PSYNC signal goes true for one clock period. For a read cycle, PDBIN is asserted in the middle of the PSYNC signal and also remains true for one clock period.

For a write cycle, the PWR signal has been shortened to one half cycle in length, true at the end of PSYNC. This may cause problems with some memories, and, in fact, a transparent write cycle may not be possible unless the BUS AVAILABLE signal is true longer.

The BUS 2 signal is released on BUS AVAILABLE inactive and the BUS 1 signal is released on the next rising edge of Ø.

The Z-80 refresh (RFSH) signal meets the requirements for use as a BUS AVAILABLE signal.

16 BIT PROCESSORS ON THE S-100 BUS

A proposal has been arrived at that allows both existing 8 bit masters and 16 bit masters of a new design to operate in the same bus. This allows current systems to be used as development systems for the new 16 bit processors. Two lines, Sixteen Request (SXTRQ) and Sixteen Acknowledge (SXTN) have been assigned to the bus to control the ganging of the Data In and Data Out buses, and the technique requires memory boards of a unique design.

The key to the system is to allow the 16 bit processor to only assert even addresses to the bus for memory access. This does not mean that a 16 bit address space is reduced to 15 bits,but rather that the processors A0-A15 are asserted on bus A1-A16, with A0 asserted as zero.

Consider, as an example, a read operation. An even address is asserted on the address bus and the Sixteen Request line is active. The addressed memory board decodes Status and address in conjunction with Sixteen Request, and asserts Sixteen Acknowledge. The memory board places the high order byte on the DI bus and the low order byte on the DO bus during the PDBIN interval, and the read is complete. If Sixteen Acknowledge is not received by the processor, it may perform the operation in byte serial fashion, or trap to an error routine.

An 8 bit master, reading the same word from memory, asserts an even address without the Sixteen Request line, and receives the low order byte on the DI bus. By asserting the odd address without the Sixteen Request line, the processor receives the high order byte on the DI bus. The case for write operations is similar, except that the DO bus carries data written by an 8 bit master.□

SUMMARY OF DELETIONS AND ADDITIONS

Deletions:

RUN and SINGLE STEP

PROTECT, UNPROTECT, and PROTECT STATUS

INTERRUPTS ENABLED

SSTACK

ADDRESS DISABLE has been combined with DATA OUT DISABLE, freeing one line.

STATUS DISABLE has been combined with COMMAND/CONTROL DISABLE, freeing one line.

Additions:

A 16-A 23, Extended Address lines

DMARQ0-DMARQ7, Vectored DMA Request

SXTRQ, Sixteen Request

SXTN, Sixteen Acknowledge

BUS AVAILABLE, for transparent DMA

STANDBY POWER, 5 volt power independent of power switch

NMI, Non-maskable interrupt

HOLD/INTERRUPT ACKNOWLEDGE (three lines)

POWER DOWN, indicates impending power failure

REFRESH, Refresh signal for dynamic memories

Redefinitions:

CLOC, now always 2 mhz

01, now Status strobe

S-100 TIMING SPECIFICATION

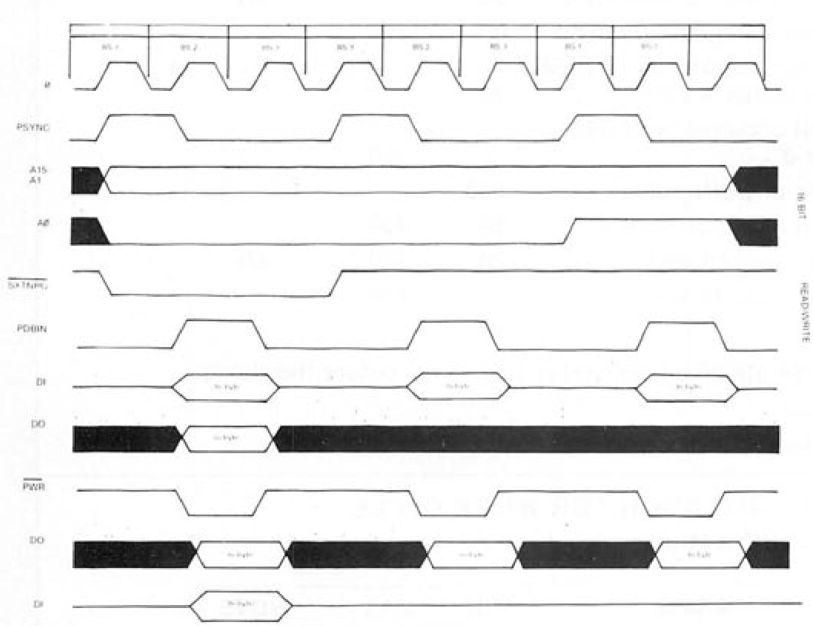


Figure 8.

All times in the following section are given in nanoseconds.

Ø is designated as the system clock. Bus states begin immediately after the falling edge of Ø, and last for one clock period.

Rise and fall times of Ø are specified as not greater than 50 ns. for a 2 mhz. clock frequency, and not greater than 30 ns. for a 4 mhz. clock frequency.

SPECIFICATION FOR READ CYCLE

CLOCK FREQUENCY: 2 Mhz.-6 Mhz.

SYMBOL	NAME	MIN	MAX	NOTE
t_{cy}	clock period	166	500	
$t_{\theta sn}$	delay from θ to sync	5	$t_{cy}/4+5$	
$\overline{t_{\theta sn}}$	delay from θ to $\overline{\text{sync}}$	5	$t_{cy}/4+5$	
t_{as}	address stable before ststb	20	---	
t_{sts}	status stable before ststb	20	---	
t_{snst}	status changes after sync	0	$t_{cy}/2$	
t_{stdb}	earliest occurrence of DBIN after status stable	30	---	
$t_{\theta db}$	latest occurrence of DBIN after θ	---	$t_{cy}/4+25$	
t_{db}	width of DBIN pulse	t_{cy}	---	
t_s	width of status strobe	50	$t_{cy}/2$	
t_{sns}	delay sync to ststb	t_{as}	t_{cy}-t_s	
$\overline{t_{ssn}}$	delay $\overline{\text{ststb}}$ to $\overline{\text{sync}}$	0	t_{cy}-t_s	

SPECIFICATION FOR READ CYCLE

CLOCK FREQUENCY: 2 Mhz.

SYMBOL	NAME	MIN	MAX	NOTE
t_{cy}	clock period	500	---	
$t_{\theta sn}$	delay from θ to sync	5	130	
$\overline{t_{\theta sn}}$	delay from θ to $\overline{\text{sync}}$	5	130	
t_{as}	address stable before ststb	20	---	(1)
t_{sts}	status stable before ststb	20	---	(1)
t_{snst}	status changes after sync	0	250	
t_{stdb}	earliest occurrence of DBIN after status stable	30	---	
$t_{\theta db}$	latest occurrence of DBIN after θ	---	150	
t_{db}	width of DBIN pulse	500		
t_s	width of status strobe	50	250	
t_{sns}	delay sync to ststb	20	450	(2)
$\overline{t_{ssn}}$	delay $\overline{\text{ststb}}$ to $\overline{\text{sync}}$	0	450	

NOTES:

(1) Status must be stable *on the status bus* 20 ns. before the rising edge of STSTB.

(2) $t_{snst}(min) + t_{sts}$

SPECIFICATION FOR WRITE CYCLE

CLOCK FREQUENCY: 2 Mhz.

SYMBOL	NAME	MIN	MAX	NOTE
$t_{\theta\overline{wr}}$	delay from θ to $\overline{\text{wr}}$	5	500	
$t_{\overline{wr}}$	pulse width of $\overline{\text{pwr}}$	500	---	
$t_{d\overline{owr}}$	data stable before trailing edge of $\overline{\text{pwr}}$	100	---	
t_{wrdo}	data stable after trailing edge	100	---	
t_{wra}	address and status stable after trailing edge of $\overline{\text{pwr}}$	100	---	
$t_{\overline{wr}mw}$	delay from $\overline{\text{wr}}$ to mwrite	---	30	
$t_{wr\overline{mw}}$	delay from wr to $\overline{\text{mwrite}}$	---	30	

FIRST CLASS
PERMIT NO. 11
CERRITOS, CA
90701

BUSINESS REPLY MAIL

No Postage Stamp Necessary if Mailed in the United States

Postage will be paid by

INTERFACE AGE™ Magazine

P.O. Box 1234

Cerritos, CA 90701

FIRST CLASS
PERMIT NO. 11
CERRITOS, CA
90701

BUSINESS REPLY MAIL

No Postage Stamp Necessary if Mailed in the United States

Postage will be paid by

INTERFACE AGE™ Magazine

P.O. Box 1234

Cerritos, CA 90701

SPECIFICATION FOR WRITE CYCLE

CLOCK FREQUENCY: 2 Mhz.-6 Mhz.

SYMBOL	NAME	MIN	MAX	NOTE
$t_{\theta\overline{wr}}$	delay from θ to $\overline{wr}$	t_{osn}	t_{cy}	
$t_{\overline{wr}}$	pulse width of $\overline{pwr}$	t_{cy}	---	
$t_{d\overline{owr}}$	data stable before trailing edge of $\overline{pwr}$	$t_{cy}/5$	---	
t_{wrdo}	data stable after trailing edge	$t_{cy}/5$	---	
t_{wra}	address and status stable after trailing edge of $\overline{pwr}$	$t_{cy}/5$	---	
$t_{\overline{wr}mw}$	delay from $\overline{wr}$ to mwrite	---	30	
$t_{wr\overline{mw}}$	delay from wr to $\overline{mwrite}$	---	30	

SPECIFICATION FOR WAIT CYCLE

CLOCK FREQUENCY: 2 Mhz. or 4 Mhz.

SYMBOL	NAME	MIN	MAX	NOTE
$t_{\overline{r}\theta}$	setup of $\overline{rdy}$ before θ	50	---	(2)
$t_{r\theta}$	setup of rdy before θ	50	(4)	(2)
$t_{\overline{\theta}r}$	rdy held after $\overline{\theta}$	50(5)	---	(2)
$t_{\overline{\theta}wt}$	pwait true after θ	0	---	(1)
$t_{wt\theta}$	pwait true before θ	50		(1)(3)
$t_{\overline{wt}\theta}$	pwait false before θ	---	---	(1)

NOTES:

(1) Should reflect processor sample of wait request.

(2) $\overline{rdy}$ and rdy must be true for both the rising and falling edges of θ., unless (4)or (5).

(3) MIN must include time for response release.

(4) rdy must not extend into the previous θ = TRUE period unless reset by pwait = TRUE

(5) rdy must be true until pwait = FALSE

SPECIFICATION FOR DMA CYCLE

CLOCK FREQUENCY: 2 Mhz.-6 Mhz.

SYMBOL	NAME	MIN	MAX	NOTE
$t_{ha\theta}$	HLDA setup before θ	$t_{cy}/5$		
$t_{\overline{\theta ha}}$	delay $\overline{\theta}$ to $\overline{HLDA}$	0	$t_{cy}/2$	
$t_{\theta c}$	delay at state change	0	20	
$t_{\overline{\theta c}}$	delay at state change	0	20	
t_{b2ph}	delay BUS 2 to PHOLD high	0	$t_{cy}/4$	
$t_{\theta bl}$	delay θ to bus 1 low	0	$t_{cy}/2 + 50$	

NOTES:

The signal BUS 1 is comprised of $\overline{\text{ADDDSB}}$, $\overline{\text{DODSB}}$, $\overline{\text{DMA C/C EN}}$

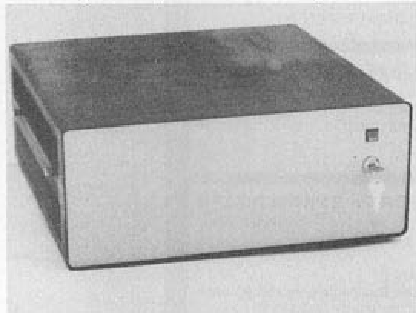
The signal BUS 2 is comprised of $\overline{\text{C/C DSB}}$, $\overline{\text{STATDSB}}$, $\overline{\text{DMA ADD EN}}$ $\overline{\text{DMA STAT EN}}$, and $\overline{\text{DM DO EN}}$

NEW PRODUCTS

Mainframe By No Name

This carefully planned and precision made mainframe that doesn't dictate to the end user what motherboard should be used.

The No Name Mainframe is engineered to accommodate S-100 compatible, off-the-shelf motherboards, i.e. Altair, Imsai, Wonderbus, and many others plus custom design boards.



No Name's assembled and tested DC power supply provides all necessary S-100 power. Also included is a fused, switched-accessory power receptacle, whisperfan, and line cord.

The rear panel accommodates 6 pre-punched DB-15 I/O connectors and 3 BNC (video) connectors. The light-beige front panel houses a lighted reset button, a keyed power switch, and is framed by a warm-brown, baked-enamel cabinet. The four knurled knobs permit easy removal of cover.

Fully assembled. Not a kit. Delivery immediate. Price $310.00. For more information contact No Name Computer Co., 5620 A East 2nd St., Long Beach, CA 90803, (213) 439-3761.

CIRCLE INQUIRY NO. 150

Personal Computing Cassettes

The Personal Computing Cassette contains 10 minutes of top quality "Scotch" brand tape to provide 5 minutes of recording time per side. this convenient length minimizes rewind time and provides sufficient storage for programs.

The Personal Computing Cassette shell is a professional 5 screw design with roller guides for accurate tape movement to minimize skew. The pressure pad is spring-loaded to provide uniform tape/head contact and minimize flutter. This outstanding shell is loaded with Scotch brand 8210 High Output/Low Noise tape with the exclusive "POSI-TRAK" backing to provide uniform wind and improved cassette mechanical performance.

Alpha Personal Computing Cassettes are available in bulk at $1.00 each, in a plastic hinged box at $1.30 each or the unique "C-Box" storage unit at $1.70 each. For more information or to order contact Alpha Supply Co., 18350 Blackhawk St., Northridge, CA 91326. Dealer inquiries invited.

CIRCLE INQUIRY NO. 181

Latest Series 21 Addition from MDS

Mohawk Data Sciences has a new distributed processing system that doubles the performance range of its Series 21™ product family.

The new System 21/50 is hardware and software comparible with System 21/40, previously the high end of Series 21, yet offers twice the processing speed, twice the main memory, and twice the local file storage capacity of either System 21/40 or the entry level System 21/20.

System 21/50 features a central processor unit twice as fast as those used on the earlier systems. It also uses multiple microprocessors and an interlaced memory system.

System 21/50 also brings multiprogramming and concurrent task execution to Series 21. A unique "virtual" keystation allows the system to handle up to three concurrent tasks, including one "background" task such as batch communication with a host computer or media conversion.

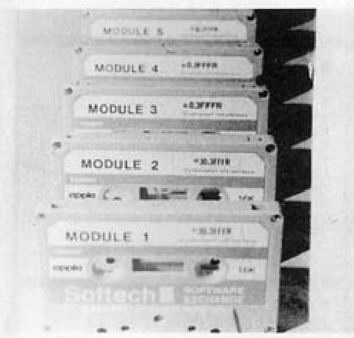
A four-station System 21/50 with 128K bytes of memory, two diskette drives, 20-megabyte fixed disk, 340 lines-per-minute printer, 25 inches-per-second magnetic tape drive, and communications interface, is priced at $1,225 per month on a 3-year lease plus $343 per month maintenance. Selling price is $58,869.

For more information contact Mohawk Data Sciences Corp., 1599 Littleton Rd., Parsippany, NJ 07054.

CIRCLE INQUIRY NO. 156

Softape Software Exchange

The Softape Software Exchange was created to interface the microcomputer owner and the microcomputer programmer. Now thqusands of programs will be at our fingertips. Every kind of program will be available quickly and very inexpensively.



The Exchange presently has an extensive library for the Apple II computer, systems using North Star Disk BASIC or Tarbell cassette interface and Radio Shack TRS-80.

Softape is actively seeking any new software for personal computers to be distributed in the Exchange. Members will receive credit toward their membership fee if their software is published in the exchange. Softape is not based on any product line and we will support all personal microcomputers.

The largest problem in personal computing today is the lack of organization and distribution of software. Much software exists but is not readily obtainable. Softape is committed to filing this void. Since you had the insight to join the microcomputer revolution, we have no doube that you recognize the value of this opportunity. After all — the Softape Software Exchange is as strong as its members, and limited only by their creativity.

For more information contact Softape Software Exchange, 10756 Vanowen St., No. Hollywood, CA 91605. The Softape Software Exchange is not affiliated with Softech Inc. in Massachusetts.

CIRCLE INQUIRY NO. 201

6800 CPU for S-100

A 6800-based, single board computer for the S-100 bus has PDP-11-like instruction set. Turn-key operation makes building a business system both easy and inexpensive.

Included on the board are an RS-232/20mA TTY I/O port (ready to hook up to your terminal), a 1K byte Operating System in ROM, Scratchpad RAM, slow memory and dynamic memory interfacing, hardware single-step, and power on restart.

CCI

VACATION BOOK SALE

Basic Software Library—Vol. I & II	$24.95
Basic Software Library—Vol. III & VII	$39.95
Basic Software Library—Vol. IV & V	$ 9.95
Basic Software Library—Vol. VI	$49.95
Basic Software Library—Vol. VIII	$19.95
Creative Computing—Vol. I & II	$ 8.95
Instant Basic	$ 7.50
101 Basic Games	$ 7.50
8080 Bug Book	$ 9.95
Sam Z-80 Program	$ 8.50
Cheap Video Cook Book	$ 5.95
Basic Basic	$ 8.95
Advanced Basic	$ 6.95
I.C. Timer	$ 9.95
TTL Cook Book	$ 9.50
Your Own Computer	$ 1.95
Z-80 Assembly Language	$ 7.50
Z-80 ZPU Technical Manual	$ 7.50
Z-80 Program Logic Design	$ 7.50
First Book of Kim	$ 8.95
SYBEX Microprocessor	$ 9.95
SYBEX Micro Interfacing Techniques	$ 9.95
How to Build a Computer Controlled Robot	$ 7.95
OSBORNE	
Intro to Micro—Vol. 0 & I	$ 7.50
Intro to Micro—Vol. II	$15.00
8080 Programming	$ 7.50
6800 Programming	$ 7.50
280 Programming	$ 7.50
Some Common Basic Programs	$ 7.50
Payroll with Cost Accounting	$12.50
C-MOS Cook Book	$10.50
How to Program Micro Computer	$ 8.50
BUG BOOKS	
Bug Book Vol. I & II & IV & VII	$ 8.50
Bug Book Vol. III	$15.00
Bug Book Vol. V & VI	$ 9.50
The 555 Timer Applications	$ 6.95
The Design of Active Filters	$ 8.50

....

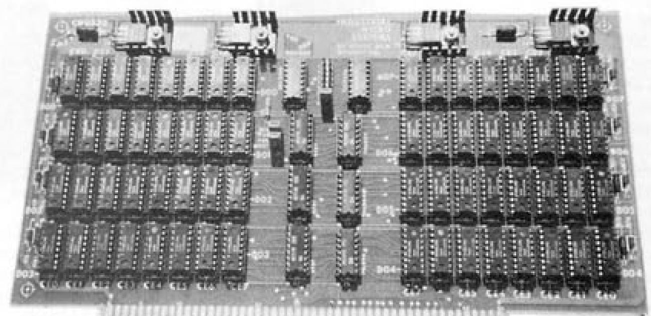
WE ALSO STOCK A COMPLETE LINE OF THE FOLLOWING BOOKS:

SAMS	TAB
HAYDEN	PRENTICE HALL
WILEY	DILITHIUM PRESS

ALL BOOKS 10% OFF
TAB BOOKS 20% OFF

PRICES IN EFFECT THRU SEPT. 30

IMS
MEMORY MADNESS



INDUSTRIAL MICRO SYSTEMS

8K STATIC

GUARANTEED TO RUN AT 4 Mz

$159.95

"WORLDS FINEST 8K MEMORY MODULE"

....

ALSO

16K 250ns	REG. $555	**NOW $469**
16K 450ns	REG. $525	**NOW $429**

....

WE ARE A FULL LINE IMS DISTRIBUTOR

....

BANK OF AMERICA AND MASTERCHARGE WELCOME. TERMS: MIN. ORDER $10.00 ADD $2.00 POSTAGE AND HANDLING IF ORDER IS UNDER $25.00 AND SENT U.P.S. ADD $4.00 POSTAGE AND HANDLING IF SENT VIA U.S. MAIL.

CCI

COMPUTER COMPONENTS INC.

5848 Sepulveda Blvd., Van Nuys, CA 91411
(213) 786-7411
4705 Artesia Blvd., Lawndale, CA 90260
(213) 370-4842
CLOSED SUNDAYS AND MONDAYS

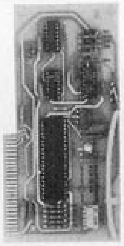
COMPUTER INTERFACES & PERIPHERALS

SAVE

For free catalog including parts lists and schematics, send a self-addressed stamped envelope.

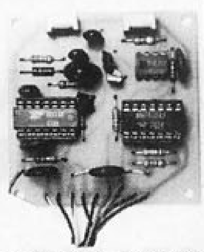
SAVE

APPLE II SERIAL I/O INTERFACE *



Part no. 2
Baud rate is continuously adjustable from 0 to 30,000 • Plugs into any peripheral connector • Low current drain. RS-232 input and output • On board switch selectable 5 to 8 data bits, 1 or 2 stop bits, and parity or no parity either odd or even • Jumper selectable address • SOFTWARE • Input and Output routine from monitor or BASIC to teletype or other serial printer. • Program for using an Apple II for a video or an intelligent terminal. Also can output in correspondence code to interface with some selectrics. Board only — $15.00; with parts — $42.00; assembled and tested — $62.00.

MODEM *

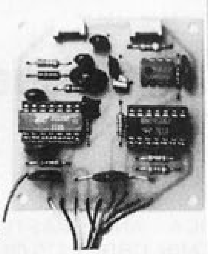


Part no. 109
• Type 103 • Full or half duplex • Works up to 300 baud • Originate or Answer • No coils, only low cost components • TTL input and output-serial • Connect 8 ohm speaker and crystal mic. directly to board • Uses XR FSK demodulator • Requires +5 volts • Board $7.60; with parts $27.50

DC POWER SUPPLY *

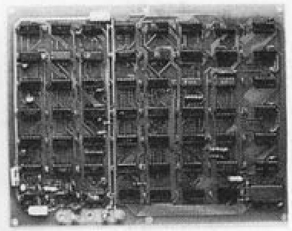
Part no. 6085
• Board supplies a regulated +5 volts at 3 amps., +12, -12, and -5 volts at 1 amp. • Power required is 8 volts AC at 3 amps., and 24 volts AC C.T. at 1.5 amps. • Board only $12.50; with parts excluding transformers $42.50

TAPE INTERFACE *



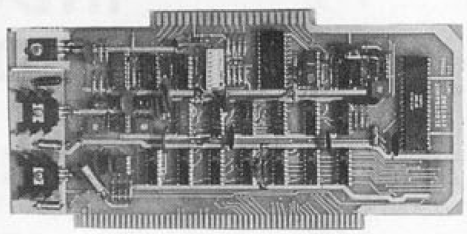
Part no. 111
• Play and record Kansas City Standard tapes • Converts a low cost tape recorder to a digital recorder • Works up to 1200 baud • Digital in and out are TTL-serial • Output of board connects to mic. in of recorder • Earphone of recorder connects to input on board • No coils • Requires +5 volts, low power drain • Board $7.60; with parts $27.50

T.V. TYPEWRITER



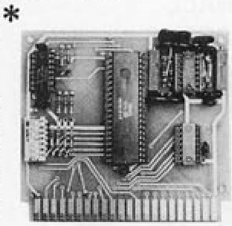
Part no. 106
• Stand alone TVT • 32 char/line, 16 lines, modifications for 64 char/line included • Parallel ASCII (TTL) input • Video output • 1K on board memory • Output for computer controlled curser • Auto scroll • Non-destructive curser • Curser inputs: up, down, left, right, home, EOL, EOS • Scroll up, down • Requires +5 volts at 1.5 amps, and -12 volts at 30 mA • All 7400, TTL chips • Char. gen. 2513 • Upper case only • Board only $39.00; with parts $145.00

TIDMA *



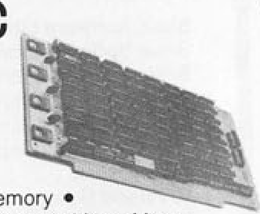
Part no. 112
• Tape Interface Direct Memory Access • Record and play programs without bootstrap loader (no prom) has FSK encoder/decoder for direct connections to low cost recorder at 1200 baud rate, and direct connections for inputs and outputs to a digital recorder at any baud rate. • S-100 bus compatible • Board only $35.00; with parts $110.00

UART & BAUD RATE GENERATOR*



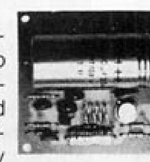
Part no. 101
• Converts serial to parallel and parallel to serial • Low cost on board baud rate generator • Baud rates: 110, 150, 300, 600, 1200, and 2400 • Low power drain +5 volts and -12 volts required • TTL compatible • All characters contain a start bit, 5 to 8 data bits, 1 or 2 stop bits, and either odd or even parity. • All connections go to a 44 pin gold plated edge connector • Board only $12.00; with parts $35.00 with connector add $3.00

8K STATIC RAM



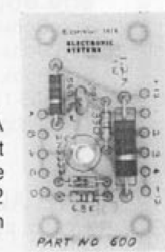
Part no. 300
• 8K Altair bus memory • Uses 2102 Static memory chips • Memory protect • Gold contacts • Wait states • On board regulator • S-100 bus compatible • Vector input option • TRI state buffered • Board only $22.50; with parts $160.00

RF MODULATOR *



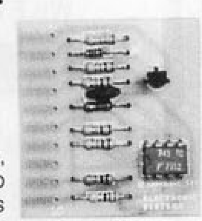
Part no. 107
• Converts video to AM modulated RF, Channels 2 or 3. So powerful almost no tuning is required. On board regulated power supply makes this extremely stable. Rated very highly in Doctor Dobbs' Journal. Recommended by Apple. • Power required is 12 volts AC C.T., or +5 volts DC • Board $7.60; with parts $13.50

RS 232/TTY * INTERFACE



Part no. 600
• Converts RS-232 to 20mA current loop, and 20mA current loop to RS-232 • Two separate circuits • Requires +12 and -12 volts • Board only $4.50, with parts $7.00

RS 232/TTL* INTERFACE



Part no. 232
• Converts TTL to RS-232, and converts RS-232 to TTL • Two separate circuits • Requires -12 and +12 volts • All connections go to a 10 pin gold plated edge connector • Board only $4.50; with parts $7.00 with connector add $2.00

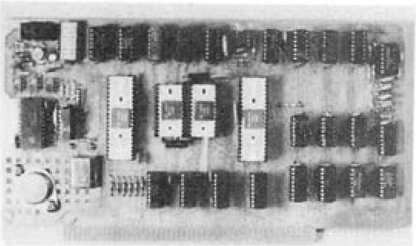
ELECTRONIC SYSTEMS

Dept. IA, P.O. Box 21638, San Jose, CA. USA 95157

To Order: Mention part number and description. For parts kits add "A" to part number. In USA, shipping paid for orders accompanied by check, money order, or Master Charge, BankAmericard, or VISA number, expiration date and signature. Shipping charges added to C.O.D. orders. California residents add 6.5% for tax. Outside USA add 10% for air mail postage, no C.O.D.'s. Checks and money orders must be payable in US dollars. Parts kits include sockets for all ICs, components, and circuit board. Documentation is included with all products. All items are in stock, and will be shipped the day order is received via first class mail. Prices are in US dollars. No open accounts. To eliminate tariff in Canada boxes are marked "Computer Parts." Dealer inquiries invited.
24 Hour Order Line: (408) 226-4064

VISA master charge

* Circuits designed by John Bell



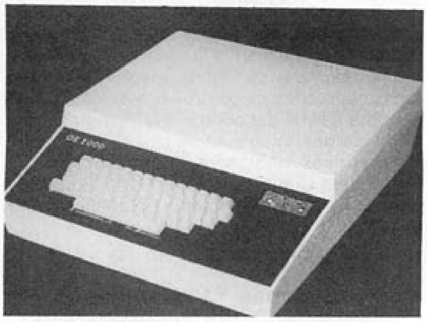
Operating Systems in ROM which are currently available offer many software debug utility routines, real-time operation, and multitask programming.

Complete assembled Business Systems start at $5,999. The 6800 CPU board alone sells for $179 in kit form, and $269 assembled. For more information contact Datatronics, Inc., 208 E. Olive, Lamar, CO 81052, (303) 336-7956.

CIRCLE INQUIRY NO. 152

OE 1000 Terminal

The OE 1000 terminal is designed to interface to any microcomputer that has a 300 baud serial data output port. It operates in the full duplex mode with either 20 ms current loop or a RS-232 interface.



The OE 1000 outputs composite video for use with a modified TV or video monitor. The screen format is 16 lines by 64 characters. It has an upper and lower case mode or TTY mode keyboard and will display 96 ASCII characters and 32 special characters. The OE 1000 has full cursor control, automatic scroll, erase to end of line, erase to end of screen and clear screen.

The OE 1000 terminal is $275 for kit or $350 assembled. Delivery is 2 weeks. Dealer inquiries invited. For more information or direct ordering contact Otto Electronics, P.O. Box 3066, Princeton, NJ 08540.

CIRCLE INQUIRY NO. 168

SWTP Prototype Boards

Celetron has available two different prototype boards for use with the SWTP 6800 computer system. They are intended for use with wire-wrap sockets.

The smaller board is identified as the SWT-1 and is intended to plug into the I/O portion of the SWTP bus; it is priced at $8.95. The SWT-2 is a larger board and plugs into the memory bus. Molex connectors for plugging the boards into the system bus are included.

The boards may be purchased through computer stores, or may be ordered from Celetron direct if a $1.50 per order is included to cover shipping, handling and insurance.

For more information contact Celetron, P.O. Box 6215, Syracuse, NY 13217, (315) 422-6666.

CIRCLE INQUIRY NO. 175

SuperPac 180 Development Systems

The SuperPac 180 Development System is designed to support the development and implementation of 180 Microcomputer Systems. The basic SPDS includes a dual floppy disk, 1806 microcomputer module (8080A CPU, 1K bytes of RAM, sockets for 7K bytes of 2708, 8708/2308, EPROM/ROM memory (5K bytes of EPROM included programmed with SPDOS), 8

vectored priority interrupts, 5 interval timers, 20ma serial port, 8 TTL DI's, 8 TTL DO's), 1813 ROM/RAM memory module (16K bytes of RAM, swtich selectable "WRITE PROTECT", sockets for 16K bytes of 2708, 8708/2308 EPROM/ROM memory), 1823 TTL I/O module (64 TTL DI's, 64 TTL DO's), CH20 twenty-slot card chassis and PS300, 300 watt switching power supply.

For more information contact Process Computer Systems, Inc., 750 N. Maple Rd., Saline, MI 48176, (313) 429-4971.

CIRCLE INQUIRY NO. 159

Conductive Rubber Floor Mats

The new "Conducto-Mat" conductive rubber floor mat is designed for use with minicomputers, microcomputers, computer terminals, word processors, cash registers or other electrical equipment containing microcircuits that

can be damaged by static electricity generated by the operator.

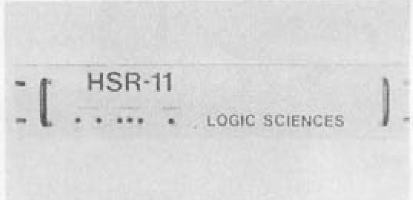
The standard 3x8-foot "Conducto-Mat" is available with a *built-in* snap fastener for attaching a ground strap. This is necessary because a carpet of synthetic materials prevents the mat from dissipating static electricity.

The conductive rubber mat is also available in rolls for use as floor runners in areas of heavy foot traffic. For more information contact Wescorp, 1155 Terra Bella Ave., Mountain View, CA 94040.

CIRCLE INQUIRY NO. 200

HSR-11 Controller

The HSR-11 is a new system for high-speed seismic wiggle-trace raster generation. The unit interfaces between a host processing system and an electro-static printer/plotter. It converts seismic trace sequential data to raster



scan lines and drived the attached printer/plotter.

The HSR-11 interfaces to any of the standard electrostatic plotters. In addition to seismic wiggle-trace generation, the HSR-11 features vector generation from X, Y end point defined vectors; alphanumeric character generation, rotation, and scaling; seismic trace overlap; user-defined fill options; automatic timing line generation; and a memory expansion feature.

Delivery time is from 30 to 45 days for standard configurations. For more information contact Logic Sciences, Inc., 6440 Hillcroft, Suite 412, Houston, TX 77081, (713) 777-8744.

CIRCLE INQUIRY NO. 154

6802 and 8085-Based Development Systems

Futuredata offers a range of disk-based microsystems for the 6802 and 8085 to allow the designer to select the level of performance required. Each includes the microprocessor CPU with up to 64K of memory, high speed 960 character CRT, ASCII keyboard, dual floppy disk or cassette tape unit, operating system software and documentation.

System features include two RS-232 serial ports, 8-bit parallel TTL I/O port, real-time clock, bootstrap in PROM, memory write-protect under software control, 8-level vectored interrupts, DMA capability and complete disk and tape operating systems.

For prices and more information contact Futuredata Computer Corp., 11205 S. La Cienega Blvd., Los Angeles, CA 90045, (213) 641-7700.

CIRCLE INQUIRY NO. 153

Synapse™10 Handles Communication Protocols and Industrial Control

The Synapse/10 is an 8080-based microcomputer for data acquisition, industrial control, terminal buffering, and communications interfacing between devices with different protocols.

Its memory consists of up to 8K RAM and 8K EPROM. The system contains 48 parallel I/O lines, two independent synchronous or asynchronous serial ports, priority interrupts, real time clock, and fan-cooled 30 watt power supply in a self-contained metal enclosure. Software development is available for all of the above applications.

For more information contact Morrow Computer & Electronic Design, Inc., 315 Wilhagan Rd., Nashville, TN 37217.

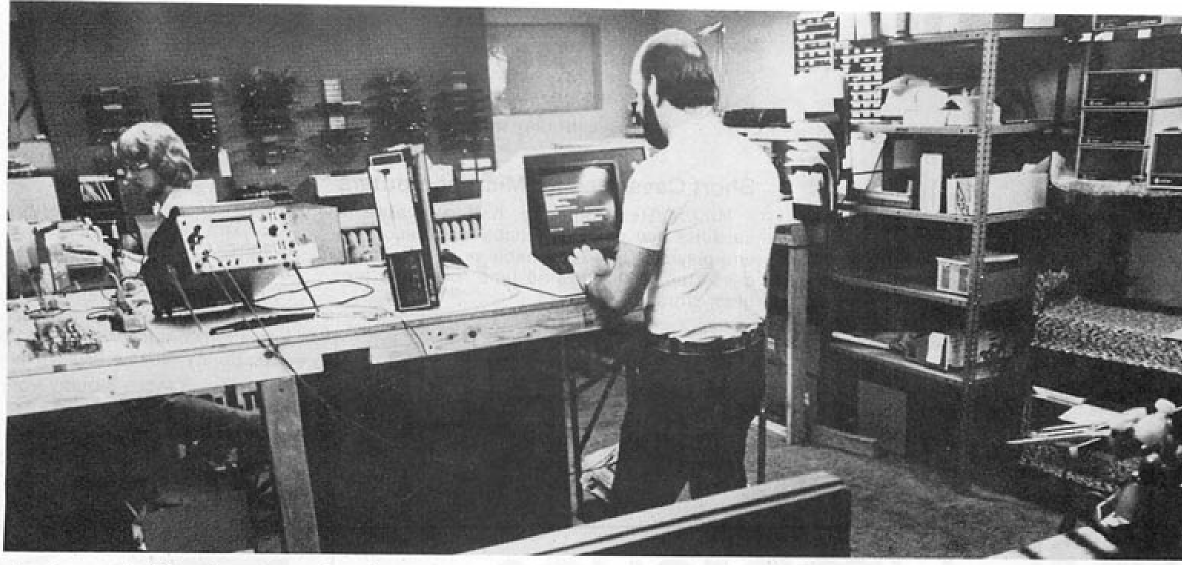
CIRCLE INQUIRY NO. 155

Interactive Transaction System

TRAX™ is a comprehensive minicomputer-based system for interactive transaction processing. The new system is priced significantly lower than other systems of any size with comparable capability.

The PDP-11-based system is designed to manage the collection, organization, storage and retrieval of business transaction information for financial institutions, insurance companies, government agencies, manufacturers

CIRCLE INQUIRY NO. 83



Just a part of our large service department . . .

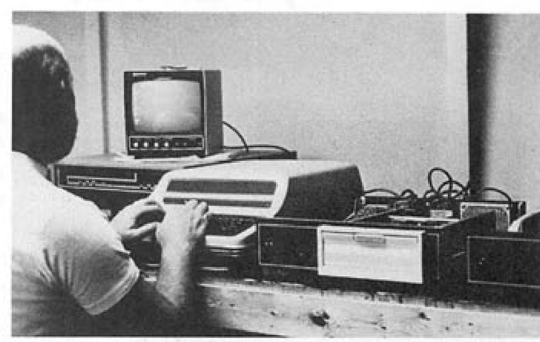
Our service department alone is bigger than some computer stores.

Bigness doesn't always mean better. But with nearly 1,000 square feet of space for our service department alone, you know we're serious about offering you support. And with 4 full-time technicians, you know we've committed more than just space to giving you quality service.

Service that's as tailored to your needs as your system is to your particular application. Priority "while-you-wait" bench service for those of you who need your computers right away. Normal bench service for you who can wait 2 to 3 weeks for your machines. Full-coverage maintenance agreements, especially suited to our business customers, with factory-authorized modifications, preventive maintenance, breakdown service, and supply delivery included in the monthly fee.

But offering you service **after** a sale is only part of what we do. All systems sold by The Computer Store of Michigan, Inc. are given a 3-day burn-in **before** they ever leave the store. Even if the system came assembled and tested from the factory. We try to prevent problems, not just solve them.

That's why we welcome your questions. As an authorized MITS dealer, we can answer most any question about the Altair line of microcomputers.



Every system goes through a 3-day diagnostic burn-in (shown here) before it ever leaves the store.

And since our salespeople are all computer professionals, many other hardware and software questions can be answered just by talking with one of them.

So come to the Computer Store of Michigan, Inc., where we support every sale we make. With service. With knowledge. And most important of all, with experience.

Michigan's only authorized MITS® dealer

The Computer Store of Michigan, Inc.

The Computer Store of Ann Arbor
310 E. Washington St.
Ann Arbor, MI. 48104
(313) 995-7616

The Computer Store of Detroit, Inc.
3290 E. South Blvd.
Bloomfield Hills, MI. 48013
(313) 852-6411

The Computer Store of Macomb, Inc.
42946 Garfield
Mt. Clemens, MI. 48044
(313) 286-6700

The Computer Store of Toledo, Inc.
8 Hillwyck
Toledo, Ohio 43615
(419) 535-1541

mits®
Products of Pertec Computer Corporation

CIRCLE INQUIRY NO. 69

and other commercial and industrial users.

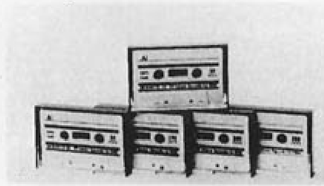
The multiterminal, high-performance system runs on PDP-11/34, PDP-11/60 and PDP-11/70 computers and uses a new video display application terminal, the VT62.

Packaged PDP-11/34 TRAX systems begin at $114,990; PDP-11/70 systems are priced from $141,620. For more information contact Digital Equipment Corp., Maynard, MA 01754, (603) 884-5101, Joseph C. Nahil.

CIRCLE INQUIRY NO. 151

Short Cassettes for Microcomputers

Microsette Company has available a cassette that provides reliable data recording and playback at a reasonable price. The short cassettes reduce rewind time for single program storage.



The high energy tape used in Microsette Data Tapes has been selected for consistency *in output envelope* and is *error-free* for the *recording* densities used by all popular home and hobby computers. The products ar backed by a user-oriented warranty covering defects in materials or workmanship.

Each cassette comes with a hard box and two extra sets of labels. Prices for each C-10, C-20, C-40 and C-60 length cassette respectively is $0.65, $0.75, $0.90 and $1.00. A sample 50-foot (C-10 length) cassette is available. Send $1.00 for shipping and handling. For more information contact Microsette Co., 777 Palomar Ave., Sunnyvale, CA 94086, (408) 735-8821.

CIRCLE INQUIRY NO. 187

Custom Wood Computer Desk

A custom built all wood desk features a split level 55"x26" top with *walnut grain formica.* Upper level is 26"x26" and is perfect for a printer. Lower level is 30"x26" and places keyboard at elbow height for minimum fatigue.

Under the printer area is a 23"x23"x24" compartment with two fully adjustable shelves. Plenty of room for a CPU and disk memory. Door opening is 21"x23". Standard finish is walnut stain.

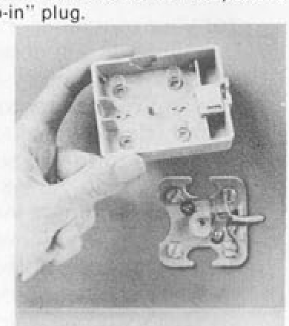
Options available are right or left hand design, other finishes and tops, and minor variations in dimension sizes. Desks are built to customer's requirements so send 50% deposit with order. Two to six week delivery.

Desk basic price is $295. Visa and Master Charge welcome. Stephen Moe Co., P.O. Box 595, Springfield, OR 97477, (503) 726-7613.

CIRCLE INQUIRY NO. 188

42A Block Cover Permits Home Installation of "Store" Phones

This new modular connecting block cover is now available and is easily attached to a conventional 42A Block and accepts the modular "snap-in" plug.



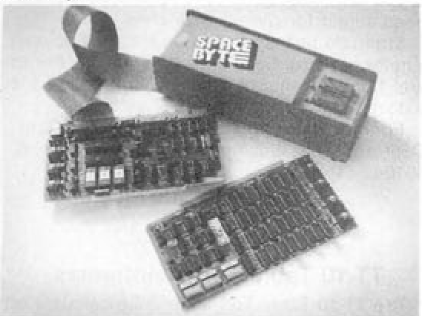
The 42A Block cover is keyed to align itself correctly over the conventional 42A Block and holds the contact springs firmly against the terminal screws of the block. The modular connecting block cover is attached with one small screw. Phone installation then requires nothing more than snapping the phone plug into the cover.

The 42A Block Cover is molded of sturdy SEO grade material with Communications Systems approved contact springs and has a captive mounting screw. For more information contact Comfast Products, Fastex Div., Illinois Tool Works, Inc., 195 Algonquin Rd., Des Plaines, IL 60016.

CIRCLE INQUIRY NO. 185

Space Byte 8085 CPU

The Space Byte 8085 CPU is a single card, self-contained computer developed specifically for the small business system packaged at the retail level. The Space Byte 8085 CPU, on one S-100 card, operates at 3MHz, using 450ns memory.



The Space Byte 8085 CPU is ideal for the small business computer system because of its full on-board I/O capability. There are two RS-232C serial I/O ports, with software selectable baud rates, one connects to a CRT, the other to a printer. There is a 22-bit parallel I/O port which interfaces directly with the iCOM 3700 series or Frugal Floppy Disk system.

For more information contact Space Byte Corp., 1720 Pontius Ave., Suite 201, Los Angeles, CA 90025, (213) 468-8080.

CIRCLE INQUIRY NO. 161

Four-Disk Microsystem

The Sol System IV is an integrated small computer system with four full-size floppy disks on-line. The new system includes the Sol-20 mainframe with 50,176 8-bit words of RAM memory, a Helios II Model 4 Disk Memory System, PTDOS Disk Operating System, Extended Disk BASIC, a video monitor and complete documentation. Total mass storage cap-

ability on four formatted disks is 1.5 million bytes.

Suggested domestic price for Sol System IV fully assembled and factory tested is $7995. Delivery from Sol computer dealers is stock to 90 days. For complete information contact Processor Technology Corp., 7100 Johnson Industrial Dr., Pleasanton, CA 94566.

CIRCLE INQUIRY NO. 160

Premium Quality Equipment Cabinets

Intra-Fab OEM-quality cabinets and racks are now available to computer hobbyists nationwide. They are ideally suited for the needs of the serious computer hobbyist.

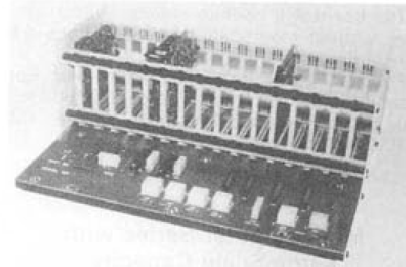
The cabinet and rack line is offered in a wide range of styles and sizes so the builder can select the enclosure best suited to the project, without making unnecessary aesthetic sacrifices.

For more information or catalog contact Intra-Fab, 660 Lenfest Rd., San Jose, CA 95133.

CIRCLE INQUIRY NO. 186

Concept 80 Microcomputer System

The CONCEPT 80 is a completely modular 8-bit microcomputer system designed specifically for demanding industrial control operations. The Concept 80 features a foiled motherboard which not only handles card to card connections but also simplifies I/O connections through standard cable connectors mounted directly on the motherboard.



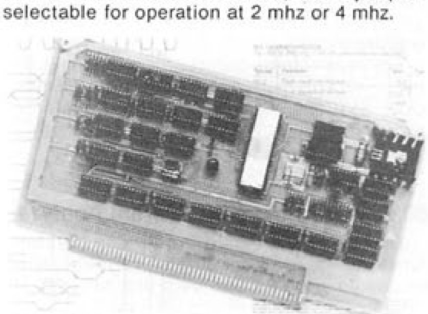
The Concept 80 employs Intel 8080 or Zilog Z-80 processor chips and is supported by software including a floppy disk operating system and a FORTRAN compiler. A full range of memory and I/O modules are available to enable the user to configure a system to exactly suit his application.

For more information contact Warner & Swasey, Computer Div., 7413 Washington Ave., Minneapolis, MN 55435, (612) 941-1300.

CIRCLE INQUIRY NO. 163

Z-80 CPU Board

This Z-80 CPU board is available assembled or in kit form and offers fully blocked design with on-board wait state select, and is jumper-selectable for operation at 2 mhz or 4 mhz.



CIRCLE INQUIRY NO. 61

The board will operate standard 8080 software without modification. All Z-80 lines are fully buffered.

Price is $175 for the kit, $215 assembled. For more information contact Vector Graphic Inc., 790 Hampshire Rd., Westlake Village, CA 91361, (805) 497-6853.

CIRCLE INQUIRY NO. 162

Minicomputer Series with Large-Scale Capacity

Called the NCSS 3200 series, these 32-byte, virtual memory systems come in five field-upgradeable models with prices ranging from $200,000 to $600,000. They will be available for sale or lease, as stand-alone systems or connected to the NCSS international data network.

Also available for the 3200 are a large number of application programs and systems, including NOMAD, a powerful proprietary data base management system. NOMAD also serves as a programming language, permitting swift development of application programs by non-computer personnel.

Augmenting the 3200 CPU and memory will be a complete selection of off-the-shelf disk and tape drives, and peripheral devices.

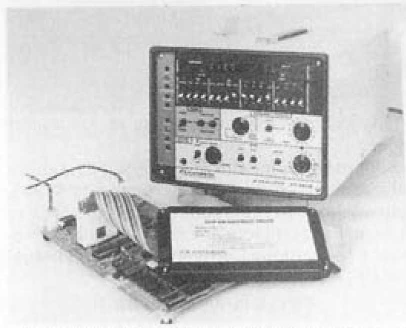
The NCSS 3200 will be marketed and supported by the National CSS Computer Division, which will also provide hardware and software maintenance.

For more information contact National CSS, Inc., 542 Westport Ave., Norwalk, CT 06851, (203) 853-7200, Bob Rushworth, Director of Communications.

CIRCLE INQUIRY NO. 192

8080 Microprocessor System Analyzer

The AQ80808 Microprocessor System Analyzer is a low cost alternative to in-circuit emulators and CRT analyzers that satisfies all diagnostic needs of 8080 family users. The AQ8080 is a cost effective, self-contained, portable instrument used for hardware development and test, program debugging, production testing, and field service.



The AQ8080 is connected directly to the microprocessor chip through a fully buffered 40-pin clip. The AQ8080 is compatible with all system configurations as it requires no memory allocation, address or I/O port assignment, special clock, or separate terminal.

Price of the AQ8080, complete with buffered probe, is $2,250. Delivery is stock to 45 days. For more information contact AQ Systems, Inc., 1736 Front St., Yorktown Heights, NY 10598, (914) 962-4264.

CIRCLE INQUIRY NO. 165

Disposable Recording Pens

Disposable recording pens in three lengths and six colors can be used on all flat-arm circular recorders. The pens slide into recorder arms in seconds. They are being used to replace V-pens, beta pens, bucket pens, pot pens and capillary tube systems.

The pens function at temperatures from −30° to +140°F and have a three-year shelf life. Each fiber-tipped pen writes 600 revolutions or 1600 feet on orifice meter charts. Before the ink is exhausted the approaching need for replacement is signalled by reduced intensity of the line.

The three lengths are .490", .770" and 1.050". Colors available in all three sizes are black, red, blue, green, brown and purple. The pens adjust for various size arms.

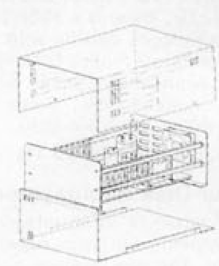
Standard packs include one dozens pens of a color per size. Case pack is six dozen. Prices are: 1 or 2 dozen, $18 a dozen; 3 to 5 dozen, $15 a dozen; 6 dozen or more, $12 a dozen.

For more information contact Sanford Corp., Dept. RB, 2740 Washington Blvd., Bellwood, IL 60104.

CIRCLE INQUIRY NO. 196

TT-10 Table Top Mainframes

The TT-10 Table Top Mainframe consists of an industrial quality Card Cage; the MB-10, an S-100 bus Mother Board with bus termination and ground plane to reduce noise; a full set of 10 connectors and guides; a 15A at 8V, 1.5A at



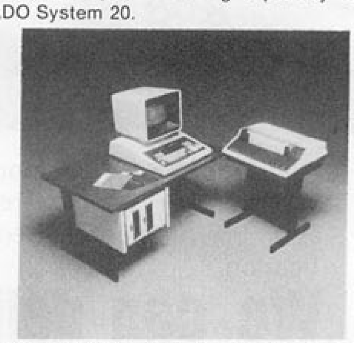
+16V and 1.5A at −16V Power Supply which mounts inside the Card Cage; a clear satin finished front/bottom plate with a reset switch and power indicator LED; a whisper fan and a vented textured blue cover.

Price is $325 for Table Top Mainframe Kit TT-10K; and $395 assembled TT-10-A. For more information contact Electronic Control Technology, 763 Ramsey Ave., Hillside, NJ 07205, (201) 686-8080.

CIRCLE INQUIRY NO. 178

Cado Adds Multi-Terminal, Multi-Tasking Capability to the System 20

The System 20/IV from Cado Systems adds multi-terminal, multi-tasking capability to its CADO System 20.

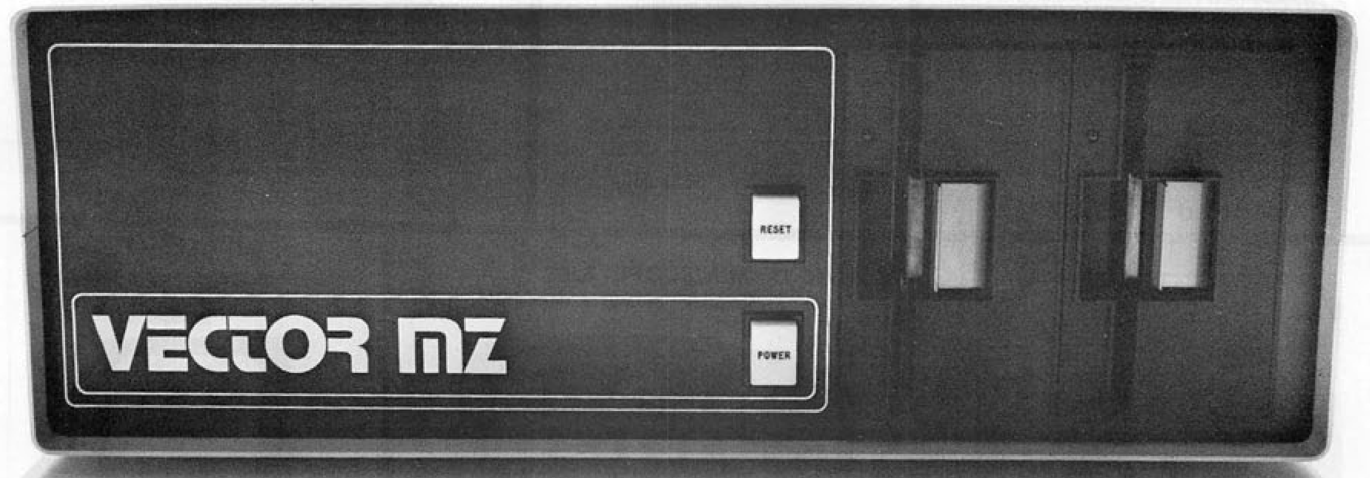


This new enhancement now offers the small business computer user a system that can have up to four video display terminals, each processing its own application independently, at an affordable cost. Cado also offers a complete series of software applications that are ready to run on the System 20/IV.

Price as low as $19,000 includes all six financial application software packages. For more information contact Cado Systems Corp., 2730 Monterey St., Torrance, CA 90503, (213) 320-9660.

CIRCLE INQUIRY NO. 164

Brand New! **VECTOR MZ**
at Super Discount Prices!



A standard MZ microcomputer, discount priced at $2695, includes these fine features: Z80 microprocessor. single floppy disk drive with 315K bytes of storage, 32K memory, one serial port, two parallel ports, disk controller, 12K PROM/RAM with monitor, Extended Disk BASIC and room for 18 S-100 boards. To expand your disk storage to 630K bytes, add another disk drive for $680.

FREE

Introductory Offer

While supply lasts.

SAVE up to $375 on your MZ System and receive FREE a BASIC programming manual and a box of diskettes.

To complete your system...

Add a SOROC Terminal and add $939. SAVE $56.00.

Add a Centronics 779 and add $1075. SAVE $100.00.

Add a Hazeltine 1500 and add $1146. SAVE $79.00.

Your total savings on a complete MZ Microsystem can add up to as much as $554.00!

All prices include full system integration, completely assembled with 90-day written warranty.

Sunshine Computer Company

20710 South Leapwood Ave. • Carson, California 90746 • (213) 327-2118

JADE

SEALS ELECTRONICS, INC.

68KSC (SWTPC 6800 compatible Static Memory Card)

- 8192 Words of Static Memory
- Access Time: 500nsec. (250 nsec on request)
- Memory Chip 91L02 APC or 2102AL-4
- Battery Standby
- Address Selected 8 ea. SPST Dip Switch
- Low Power
- All Lines Buffered.
- All IC's with Sockets
- Solder Mask on both sides of PC Board

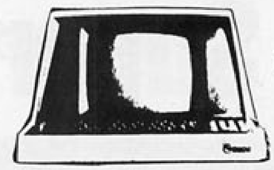
Assembled $195.00

LIQUID CRYSTAL DIGITAL CLOCK-CALENDAR

APPLIED MARKETING

- For Auto, Home, Office
- Small in size (2x2½x½)
- Push button for seconds release for date.
- Clocks mount anywhere with either 3M double-sided tape or VELCRO, included.
- 2 MODELS AVAILABLE:
  LCD-101, portable model runs on self-contained batteries for better than a year.
  LCD-102, runs on 12 Volt system and is back-lighted.
- LCD-101 or LCD-102 your choice .... **$34.95 ea.**

SOROC IQ 120 TERMINAL

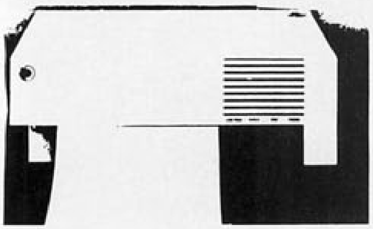


$1,025.00
Assembled

Price Includes:
- Block Mode
- Lower Case
- 24 Line Option

A CAPABLE LOW COST APPROACH TO REMOTE VIDEO DISPLAY TERMINALS.

- Cursor Control Keys Standard
- Numeric Key Pad Standard
- Line and Page Erase Standard
- Addressable Cursor Standard
- Switch Selectable Transmission from 75 to 19,200 bps Standard
- Interfaces, Printer Interface RS232 Extension
- RS232C Interface
- Non-Glare Read Out Screen
- Protect Mode Standard
- Tab Standard



Integral Data Systems INC

LOADED WITH EXTRAS AT NO EXTRA COST

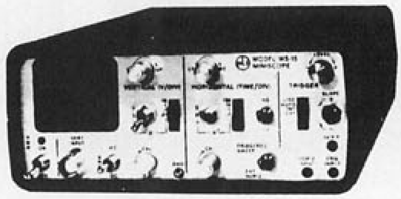
- Microprocessor controller
- Serial RS232C interface
- Parallel TTL level interface
- Full upper and lower case ASCII character set (96 characters)
- Full 8½-inch wide paper
- Line length of 80 columns at 10 characters per inch
- Impact printing
- 7 x 7 dot matrix
- Ordinary paper — roll, fanfold or sheet
- Serial baud rate to 1200 bits per second

Check the impressive features on Integral's new IP-125 Impact Printer only $799

- Multiple line buffer of 256 characters
- Built-in self test mode
- Instantaneous print rate to 100 characters per second
- Sustained throughput to 50 characters per second
- Multiple copies without adjustment
- Reinking ribbon mechanism
- Expected ribbon life of 10 million characters
- Front panel operator controls
- Attractive table top console

MS-15 MINISCOPE $289

With Rechargeable Batteries & Charger Unit



- 15 megahertz bandwidth.
- External and internal trigger.
- Time base — .1 microsec. to 0.5 Sec/div - 21 settings ± 3%.
- Battery or line operation.
- Automatic & line sync modes.
- Power consumption < 15 watts.
- Vertical Gain — .01 to 50 V/div - 12 settings ± 3%.
- Viewing area 1.1" x 1.35".
- Case size 2.7"H x 6.4"W x 7.5"D, 3 pounds.
- Parts & Labor guaranteed 1 year
- 10 to 1, 10 meg probe $24.50
- Leather carrying case $30.00

NLS

HITACHI

VIDEO MONITORS – BLACK & WHITE

VM-126
12" Solid-State Video Monitor — **352.00**
700 lines horizontal resolution
Includes black level clamping, DC restoration circuit (switchable), Ext-sync provision (Optional)

VM-909
9" Solid-State Video Monitor — **205.00**
500 lines horizontal resolution
line lock sync
High voltage 10KV

MICROPROCESSOR CRYSTALS

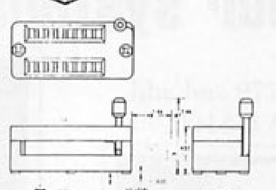
FREQUENCY	CTS P/N	price
1.0 MHz	MP010	5.95
1.8432 MHz	MP018	4.95
2.0 MHz	MP020	5.95
2.097152 MHz	MP021	8.50
2.4576 MHz	MP024	5.85
2.667 MHz	MP026	8.50
3.00 MHz	MP030	8.50
3.20 MHz	MP031	8.50
3.2768 MHz	MP032	8.50
3.579545 MHz	MP036	1.25
4.0 MHz	MP040	4.95
4.0 MHz	MP04A	4.95
4.194304 MHz	MP041	4.95
4.91520 MHz	MP042	4.95
5.0 MHz	MP05A	4.95
5.0688 MHz	MP050	4.95
5.185 MHz	MP051	4.95
5.7143 MHz	MP057	4.95
6.00 MHz	MP060	4.96
6.144 MHz	MP061	4.95
6.40 MHz	MP064	4.95
6.5536 MHz	MP065	4.95
8.0 MHz	MP080	4.95
10.0 MHz	MP100	4.95
18.00 MHz	MP180	4.95
18.432 MHz	MP184	4.95
20.0 MHz	MP200	4.95
22.1184 MHz	MP221	4.95
27.0 MHz	MP270	4.95
36.0 MHz	MP360	4.95
48.0 MHz	MP480	4.95

TEXTOOL

ZIP DIP® II Socket

This new type of zero insertion pressure dual in-line package socket (ZIP DIP II) is perfectly suited for both hand test and burn-in requirements.

The ZIP DIP II socket has been designed for the utmost simplicity in its mechanical action. Coupled with a thoughtful system of ramps and bevels to guide the device leads into the contacts results in a socket, into which, the device can literally be dropped. With the flip of a locking lever the socket is ready to operate with exceptionally good electrical contact. Flip the lever again and the device may be extracted with zero pressure being exerted on the leads by the socket contacts.



PRICES:

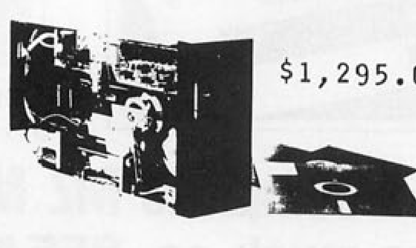
16 pin Zip Dip II	$5.50
24 pin Zip Dip II	$7.50
40 pin Zip Dip II	$10.25

ULTRAVIOLET EPROM ERASER

$34.95

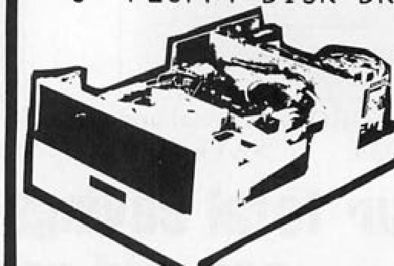
- Erases up to 48 I.C.'s at a time
- Uses popular (and readily available) 12" shortwave UV bulbs.

PERSCI MODEL 277 DUAL 8" FLOPPY DISK DRIVE



$1,295.00

SHUGART 801R 8" FLOPPY DISK DRIVE



$550.00

CRYSTAL SPECIALS

BUY 10 (MIXED) AND GET ONE $4.95 VALUE CRYSTAL FREE.....

1978 INTEL DATA BOOK
(YOUR LIBRARY ISN'T COMPLETE WITHOUT IT!)

REGULAR PRICE $3.00 — SPECIAL PRICE $2.25

VOLTAGE REGULATOR SPECIAL
340T-5(+5V)
320T-5(-5V)
340T-12(+12V)
320T-12(-12V)
75¢ EA. MIN 10

PLACE ORDERS TOLL FREE: 800/421-5809 EXCEPT CALIF.

21L02 (350ns) Static Rams 120 @ $1.00 ea.	1702A E-PROM $4.75 ea.	6502 Microprocessor 5 @ $11.00 ea.	2708 (450ns) E-PROM 8 @ $9.00 ea.	21L02 (250ns) Static Rams 100 @ $1.25 ea.	Z–80 A Microprocessor 5 @ $25.00 ea.	8212 8 Bit I/O Port 25 @ $3.00 ea.	4116 (200ns) 16K Dyn. Ram 8 @ $24.00 ea.
Z-80 Microprocessor 5 @ $20.00 ea.	8224-4 Clk.Gen.&Dvr. 25 @ $8.75 ea.	410D (200ns) Static Ram 100 @ $9.50 ea.	TMS 4044 (250ns) 16 @ $8.95 ea.	4200A (200ns) Static Rams 25 @ $10.00 ea.	74 LS367 Hex Buffer 100 @ .70¢ ea.	74 LS368 Hex Inverter 100 @ .70¢ ea.	2513 (5v) Character Gen. 5 @ $9.00 ea.

MICROCOMPUTER COMPONENTS

MICROPROCESSOR'S

F8	16.95
Z80	22.00
Z80A	28.00
CDP1802CD	19.95
2650	24.95
AM2901	22.95
6502	11.95
6800	18.95
6802	25.00
8008-1	9.95
8035	22.00
8080A	11.95
8085	27.00
TMS9900TL	75.00

8080A SUPPORT DEVICES

8212	3.50
8214	9.00
8216	3.75
8224	3.50
8224-4	9.95
8226	3.95
8228	7.95
8238	7.50
8251	9.95
8253	21.95
8255	21.95
8257	21.95
8259	21.95
8275	75.00
8279	20.00

MISC. OTHER COMPONENTS

N8T20	3.39
N8T26	2.10
N8T95	1.35
N8T96	1.35
N8T97	1.35
N8T98	1.35
81LS95	2.00
81LS97	2.00
1488	1.75
1489	1.75
D3205	4.00
D3207A	4.55
D3208A	14.20
D3211	10.00
B3222	9.75
B3242	10.15
D3245	5.60
C3404	6.75
P3408A	12.00
P4201A	5.20
MM5320	7.50
MM5369	1.90
TMS5501	24.95
DM8130	2.90
DM8131	2.75
DM8833	2.50
DM8835	2.50
DM8837	1.75
MK50240	20.00
MK50250	15.00

FLOPPY DISC CONTROLLER

1771B	55.95
1771B-01	57.95

KEYBOARD CHIPS

AY5–2376	13.75
AY5–3600	13.75

PROM'S

1702A	4.00
2704	15.00
2708	12.00
2716	30.00
2716 Intl	38.00
2758	26.60
D3601	4.50
D3604	13.00
5203AQ	5.00
5204AQ	7.50
6834	17.50
6834-1	14.95
82S23B	4.00
82S129B	4.25
8223B	3.50

6800 SUPPORT

6810P	4.95
68B10P	6.00
6820P	7.50
6821P	7.50
6828P	11.25
6834P	16.95
6850P	9.75
6852P	11.75
6860P	10.00
6862P	14.50
6871P	28.00
6875P	8.75
6880P	2.50

Z80 SUPPORT DEVICES

3881	12.95
3882	12.95

CHARACTER GENERATORS

2513	6.75
2513 5v upper	9.75
2513 5v lower	10.95
2516	10.95
MCM6571	10.95
MCM6571A	10.95
MCM6574	13.25
MCM6575	13.25

WAVEFORM GENERATOR

8038	3.50
MC4024	2.25
566	1.50

DYNAMIC RAMS

416D/4116	32.00
1103	1.00
2104	4.00
2107B	4.25
2107B-4	3.95
TMS4050	4.00
TMS4060	4.50
TMS4070-2	32.00
4096	4.00
4116/416D	32.00
MM5270	4.50
MCM6605	5.00

USRT

S2350	10.75

UART'S

AY5-1013A	5.25
AY5-1014A	8.25
TR1602B	5.25
TMS6011	5.95
IM6402	10.80
IM6403	10.80

STATIC RAMS

	1-16	17-63	64 +
21L02	1.50	1.24	1.18
21L02 (350)	1.60	1.35	1.25
21L02 (250)	1.75	1.60	1.50
410D	10.75	10.00	9.25
1101A	1.00	.90	.80
2101-1	2.95	2.75	2.60
2102	1.25	1.15	1.00
2111-1	3.95	3.50	3.25
2112-1	2.95	2.80	2.69
2114-3	11.00	10.00	9.25
2125L	11.10	9.00	8.30
2147	37.50		
31L01	2.50	2.35	2.00
3106	3.95	3.70	3.25
3107	3.95	3.70	3.25
TMS-4044	9.95	9.00	8.95
4200A	12.95		
TMS-4045	11.00	10.00	9.25
5101	8.30	7.40	7.25
74C89	3.25	3.05	2.85
7489	2.25	2.10	1.90
74S201	4.50	4.00	3.75
P8101	4.20	3.40	2.80
P8155	17.00	14.00	
P8156	21.00	18.00	
8599	1.88	1.75	1.60
9102BPC	1.65	1.45	1.30

TU-1

Convert T.V. set to Video Monitor.
KIT $8.95

JADE 8080A KIT

$100.00 KIT
BARE BOARD $30.00

MEMORY PLUS

for KIM–1
8K RAM (21L02)
8K EPROM
ASSEMBLED & TESTED
$245.00

KIM–1

ASSEMBLED & TESTED $245.00

THE APPLE II COMPUTER

One of the best "Total Package" home and business computers on the market. "Basic" in ROM, Color Graphics, Floating Point Basic Package, etc.

16K version only $1,400.00

NEC 416D 16K x 1

Dynamic Ram Chip can be used for expanding Apple II Memory:

8 for $24.00 each

full ASCII PROFESSIONAL KEYBOARDS

* Full 128 Character ASCII
* Tri-Mode MOS Encoding
* MOS DTL TTL Compatable Output
* Two-key Rollover
* Level and Pulse Strobe
* Shift and Alpha Lock
* Selectable parity
* Positive or Negative Logic.

MODEL 756 (56 keys)

PRICING INFORMATION

Model 756 (assembled)	$59.95
Model 756 K (kit)	$49.95
Model 702 enclosure	$29.95
Model 710 Numeric Pad	$9.95
Model 756 MF Mtg. Frame	$8.95

MOTHER BOARD's - S-100 STYLE

13 slot with front panel slot

BARE BOARD	$35.00
KIT	$80.00
22 Slot Assembled & Tested	$149.95

CONNECTORS

DB-25P $2.25	DB-25S $3.25
COVER $1.50	
44 Pin - PC & EYE	$1.95
44 Pin - WW	$2.50
86 Pin - (6800) PC	$5.00
86 Pin - (COSMAC ELF) PC	$5.00
100 Pin - (Altair) PC	$4.50
100 Pin - (Imsai) WW	$4.25
100 Pin - (IMSAI) PC	$3.25

MODEL 801R Shugart Disc with Cabinet

Includes Cabinet, Disc Drive, Power Supply, Cable, Fan & Data Cable. Has AC line filter.
Cabinet size 10"H x 10"W x 16"D
MODEL DM 2700-S $835.00 ea.

FLOPPY DISC INTERFACE

JADE Floppy Disc (Tarbell Board) KIT $175.00 ea.

S.D. Computer Products Versa Floppy Kit $149.00 ea.

STATIC RAM BOARDS ASSEMBLED & TESTED

8K

Ram 8 (250ns)	$169.95
Ram 8B (450ns)	$139.95
250ns KIT Mem-1	$169.95
450ns KIT Mem-1	$125.00
BARE BOARD	$25.00

16K Uses 2114L

Ram 16 (250ns)	$375.00
Ram 16B (450ns)	$325.00

32K

JG-32K (250ns)	$875.00
JG-32KB (450ns)	$775.00
450ns KIT	$675.00

6800 Adapter for S-100 system for Mem-I only KIT $12.95

16K STATIC BOARD

with memory management can be used with Alpha Micro or Cromenco Systems. ASSEMBLED & TESTED

RAM 65(250ns)	$390.00
RAM 65B (450ns)	$350.00

E-PROM BOARDS

MR-8 (8K uses 2708) KIT with 1K RAM	$99.50
MR-16T (16K uses 2716) KIT with 1K RAM	$99.50
MM-16 (16K uses 2708)	$99.00
RAM/N/ROM (16K uses any E-PROM) KIT	$117.00
JG-8/16 (uses 2708 or 2716) KIT	$59.95
BARE BOARD	$30.00

EXPANDABLE E-PROM — S.D.Sales

16K or 32K EPROM $49.95 without EPROM

Allows you to use either 2708's for 16K of Eprom or 2716's for 32K of EPROM.

COMPUTER MAINFRAME

Includes: $295.00
Power Supply +8v at 18amps ±16v at 2 amps
Mother Board - 12 slots with connectors Assembled & Tested
Has Whisper Quiet Fan & AC Line Filter
Cabinet size 7"H x 19"W x 22" D

DYNAMIC RAM BOARD

by S. D. Computer Products

On board refresh is provided with no wait states or cycle stealing required. +8VDC 400MA DC, +18VDC 400MA and -18VDC 30MA DC.

EXPANDABLE 32K (uses 4115) 200ns

8K Kit	$151.00	24K Kit	$367.00
16K Kit	$259.00	32K Kit	$425.00

EXPANDABLE 64K (uses 4116) 200ns

16K Kit	$281.00	48K Kit	$757.00
32K Kit	$519.00	64K Kit	$995.00

JADE Z80 KIT

—with PROVISIONS for ONBOARD 2708 and POWER ON JUMP

$135.00 EA. (2MHZ)

$149.95 EA. (4MHZ)

BARE BOARD $35.00

JADE VIDEO INTERFACE KIT

FEATURES $99.95

S-100 Bus compatible
32 or 64 Characters per line - 16 lines
Graphics (128 x 48 matrix)
Parallel & compositive video
On board low-power memory

Powerful software included for cursor, home, EOL, Scroll Graphics/Character Upper case, lower case and Greek. Black-on-white & White-on-black.

JADE PARALLEL/SERIAL INTERFACE KIT

$124.95 KIT

* S-100
* 2 Serial interfaces with RS232 interfaces or 1 Kansas City cassette interface.
* Serial interfaces are crystal controlled
* Selectable baud rates.
* Cassette works up to 1200 baud.
* 1 parallel port.

S-100 Power Supply with Cabinet

Power supply +8v at 18 amps ±16v at 2 amps
Has connector for Power output
MODEL UPS 600
size 5"Hx8"Wx6½"D
PRICE $149.95

JADE

Computer Products

AMERICAN EXPRESS Cards Welcome

NEW ADDRESS as of JUNE 1st

4901 W. Rosecrans
Hawthorne, Calif. 90250

master charge VISA

NEW CATALOG NOW AVAILABLE

INTRODUCTION to the

American National Standards Committee X3, for computers and Information Processing, has these active subcommittees for programming language standards.

X3J1	PL/I	X3J4	COBOL
X3J2	BASIC		
X3J3	FORTRAN	X3J6	Text Processing

Only X3J6 has a general name, not that of some specific language. Several languages exist in this area, but only one has been formally released to the subcommittee as not proprietary. It's the TEX language, developed at Honeywell Information Systems by Eric Clamons and Richard Keys, with me as advisor. My guess is that the final standard text processing language will look very much like TEX. Two reasons are: 1) it is a powerful general language that is extremely easy to use, and 2) it has the same antecedents as IBM's language SCRIPT.

I'll venture even further, predicting that TEX may supplant (or at least subsume) the BASIC language. The reason for this article is, of course, to give you an advance view of such future usage.

TEX IN GENERAL

TEX differs from many other programming languages in having the possibility of more than one active element. The program you write can act alone, operating on self-contained data, but more often it acts in conjunction with a file called the "current file", which became the current file by bringing it from the permanent store by saying:

old filename

From this point on TEX may use, in addition to its normal programming language features, the elements of a text editor — to operate on that current file just as you would manually. That is, there is always a pointer to some line of that file. It may be moved backward (as far as the beginning of file) and forward (as far as end of file).

TEX WITHOUT A CURRENT FILE

I like examples to explain programming languages. Figure 1 shows results from the program "power" of Figure 2, whose letter clues on the right identify lines for explanation. TEX doesn't need line numbers. You may use them with some difficulty if you can't break the habit. Labels, identified by "!" in the first position of the line, are better. They don't have to change if one inserts or deletes lines.

```
call power                          (puts the program into execution)

  What number?        97
  Up to what power?   30

 Exponent  Value

     1     97
     2     9409
     3     912673
     4     88529281
     5     8587340257
     6     832972004929
     7     80798284478113
     8     7837433594376961
     9     760231058654565217
    10     73742412689492826049
    11     7153014030880804126753
    12     693842360995438000295041
    13     67302709016557486028618977
    14     6528362774606076144776040769
    15     633251189136789386043275954593
    16     61425365346268570446197767595521
    17     5958260438588051333281183456765537
    18     577951262543040979328274795306257089
    19     56061272466674974994842655144706937633
    20     5437943429267472574499737549036572950401
    21     527480512638944839726474542256547576188897
    22     51165609725977649453468030598885114890323009
    23     4963064143419831996986398968091856144361331873
    24     481417221911723703707680699904910046003049191681
    25     46697470525437199259645027890776274462295771593057
    26     4529654640967408328185567705405298622842689844526529
    27     439376500173838607834000067424313966415740914919073313
    28     42619520516862344959898006540158454742326868747150111361
    29     4134093490135647461110106634395370110005706268473560802017
    30     401007068543157803727680343536350900670553508041935397795649

Task completed.
```

Figure 1. Results from "power"

```
!power out:" " in:"  What number?       " y=*in x=*in  a
n=1 in:"  Up to what power?  " limit=*in                b
out:" " out:" Exponent  Value" out:" "                  c
                                                        d
!loop out:("      ",n)['6,"      ",y n=n+1              e
if n:gt:limit out:*lf,"Task completed.",*lf return      f
y=y*x goto !loop                                        g
```

Figure 2. A TEX program called "power"

Explanation of Figure 2.

a "!power" is a label. It's not used in this program. But with many such programs in a single file called "service", one would "call service!power", to execute the file "service" from this label entry point and return without executing any other subprograms in "service".

Several commands may be on a text line, separated by spaces. "out" means print at the terminal the line following the colon. Here it's a string consisting of one space.

"in" also prints the line at the terminal, but it waits for the user to reply and press the Return key. The read-

Part One

TEX Language

By Robert W. Bemer

only variable "*in" then holds the reply. Variables "y" and "x" are created to have that value.

b The power is set to 1, and the user is asked how far he wishes to take the calculation. A new variable "limit" is created for the reply.

c Three lines are printed at the terminal — a blank line, the heading, and another blank line.

d I always precede labeled lines with a line of two spaces. Program structure is easier to use. Other than this, and material at a section labeled "!explain", few documentation aids are needed.

e A labeled loop. The interior of parenthesis pairs must be evaluated first. Five blanks are put in front of the power count "n". "]" and "[" are truncation operators (and indicate direction like arrows). The part saved/used is shown by the accent acute (´). If the accent is to the left of the truncation operator, the left part is saved, and vice versa. Here the righthand 6 characters are saved.

A string of five spaces, and the value of "y", are appended to this right-justified (aligned) string. It is printed at the terminal. The value of "n", plus one, is put into "n".

f If "n" is greater than (gt) the set variable "limit", a line is printed, consisting of a Line Feed (*lf), the closing message, and another Line Feed. "return" means go back to whatever called the program to execute — another program, or the person in manual control at a terminal.

When an *if* statement occurs, the rest of the line is executed if the condition is true. Else control goes to the next line (falls through).

g "y" times (*) "x" is put into "y". The program then goes back to the label "!loop" and continues until the limit criterion is met.

SIMPLICITY OF TEX

TEX is not complex. There are no arrays to declare, or space to reserve. When a variable is created, by assignment, its space and type are set automatically. The only data type is an ASCII string, which can serve three purposes:

- The name of a variable.
- The content of a variable.
- A procedure, if the string is executable when bracketed by substitution characters (like ALGOL 68, a value will be returned if possible).

The first two properties, name and content, are easy. To define the string "man" as the name of a variable with content "arm", we enter:

```
man="arm"
```

Many characters can serve as delimiter. Double quotes are best. "arm", as a string, can be the name of another variable, as well as the content of "man". So we can enter:

```
arm="hand" hand="finger" finger="nail"
```

Now if we type "out:man", the terminal responds by printing "arm". The underscore character is used to indicate levels of indirection. If we type in "out:___man", the terminal will print "nail".

If we also type in:

```
ape="arm"
```

and then type "out:___ape", the reply will also be "nail", because a tree was established by defining apes to have arms with the same properties as men.

The HIS processor limits indirection levels to 63, to warn against getting into a loop. It will take a very complicated database to approach this number.

Now the third property — that of being an executable procedure. We may declare a variable "cp" to have string content such as:

```
cp=\splitr:("00",var):2 cents=*r splitr:*l:1\
cp=cp,\ out:"$",*l>'^"0",*r,".",cents\
```

The variable "cp" was built in two steps; the comma is the concatenation operator. You will recognize this as a dollar-edit function. The way the procedure string is read is:

- Two zeros preface the value of the variable "var". It's then split by coming in from the right two places.
- This creates *l (lefthand part), and *r (righthand part), now assigned to the variable "cents".
- Another split from the right ensures a forced zero dollar if the value is less than a dollar.
- Then the command is given ("out:") to print a dollar sign, followed by the lefthand part with the leftmost zeros removed. That is achieved by scanning right (>) until a not (^) zero character is found, and keeping the lefthand part at that point, we would have said "'<" instead. (Scanning from the right is the same, except with "<").
- Still in the "out" command, there follows a decimal point and the cents value.

Now when "substitute" mode is turned on, the TEX sees the name "cp" delimited by substitute characters, the content of "cp" will get substituted. Being executable, a

value is returned. For some examples, if the vertical bar is made the "subs" character via the statement "subs|", here are some typical results:

```
subs |
var="0000123456" |cp|
$1234.56
var=0 |cp|
$0.00
var=69 |cp|
$0.69
```

Now we'll examine "today", another TEX program that doesn't use a current file (Figure 3).

```
!today _ 1978-01-27 author:RWBemer, 602-942-1360 a
clear * scan:*date:"-" yr=*l leap=(yr/4)*4        b
scan:*r:"-" mo=*l da=*r suf="stndrdthththth"      c
del="000031059090120151181212243273304334"        d
dy="   Mon  TuesWednes Thurs   Fri Satur   Sun"   e
m1="   January  February     March     April"     f
m1=m1,"        May       June"                     g
m2="      July    August September   October"     h
m2=m2," November  December"                        i
a=(yr+11)/4 a=(a+yr)/7 incr=*rmdr+3               j
if mo:lt:7 M=(m1']](mo*10))<'" "                  k
if mo:gt:6 M=(m2']]((mo-6)*10))<'" "              l
tempord=(del']](mo*3))['3+incr+da                 m
if mo:gt:2 if leap:eq:yr tempord=tempord+1        n
fw=tempord/7 fd=tempord-fw*7+1 ord=tempord-incr   o
day=(dy']](fd*6))['6>'^" ","day"                  p
fd=fd,(suf']](fd*2))['2                           q
a=("0",ord)['2 I=a['1 X=a']]1 suffix="th"         r
if I:eq:1 if X:ne:1 suffix="st"                   s
if I:eq:2 if X:ne:1 suffix="nd"                   t
if I:eq:3 if X:ne:1 suffix="rd"                   u
ord=ord,suffix subs |                             v
out:*lf,"Today is |day|, 19|yr| |M| |da|"         w
out:" - the |fd| day of Fiscal Week |fw|,"        x
out:" - the |ord| day of the year, and"           y
time=0 sec=*time['2 hm=*time']]5                  z
if sec:lt:25 time=hm goto !end                    aa
min=hm['2 hr=hm']]2 min=("0",min+1)['2            bb
if min:eq:60 min="00" hr=hr+1                     cc
if hr:eq:24 hr="00"                               dd
time=hr,":",min                                   ee
                                                  ff
!end out:"it is now |time|",*lf nosubs return     gg
                                                  hh
!explain out:" "                                  ii
out:"'today' gives the characteristics of"        jj
out:"the moment, including date, day of the"      kk
out:"week, ordinal day, fiscal week and day,"     ll
out:"and time." return                            mm
```

Figure 3. A TEX Program for Today

```
Today is Tuesday, 1978 August 01
 - the 2nd day of Fiscal Week 31,
 - the 213th day of the year, and
it is now 08:26
```

Figure 4. Typical Output from "Today"

Explanation of Figure 3.

a The "_" following the separator space means that the rest of the line is a comment or remark. I always put a revision date here no matter what the file system does. If need be, time-of-day can be added to be more precise as to latest version. TEX is a "find-a-bug-a-minute" language (but you make fewer mistakes). It's usual to develop programs very quickly, and so require at least hourly precision!

It's also my custom to "sign" my name as program author, as suggested by Dr. Grosch, President of the Association for Computing Machinery. I go further and give my home phone number in case my programs don't work. If so (I've only been called twice), I dial up the computer (one that I use, or else the caller's) from the HIS terminal and phone just next to mine.

b "clear *" clears (destroys and negates existence of) all variables the caller of this program may have at the time. It's not good practice to use it in general service routines.

"*date" is a ready-only variable from the computer system, with the value YY-MM-DD for the current date, in International and American Standard form.

The "scan" verb is like "split", except that it breaks a string on a given substring. Here it is the "-", which becomes the content of "*m" (*middle).

"yr=" assigns the value of "*l" (YY) to the variable "yr", if it exists. If not, it creates such a variable, and remembers that it is a numeric-valued variable.

The variable "leap" is set to the value of "yr divided by 4", without remainder, multiplied by 4. For 1978, "leap" equals "76".

c The righthand part (MM-DD) is scanned for the "-", which puts "MM" in the lefthand part (*l) and "DD" in *r. These are assigned to "mo" and "da" respectively.

A variable "suf" is created with the string content shown. It can't be used in arithmetic operations or comparisons; if tried, a diagnostic will say:

```
executing file goof line #3
which=da+suf
'stndrdthththth' is not a legal number
```

The content may look strange at first, but see how it transforms to "1st 2nd 3rd 4th 5th 6th 7th".

d A numeric string is defined. Its property is that the sequential groups of 3 are the accumulated days of the year at start of each month (ignoring leap year).

e "dy" is defined to be the sequential set of names of the days (less the string "day"), in equally spaced subsets of 6 characters each.

f Another string is defined — essentially a vector when taken in groups of 10 characters.

g "m1" grows by adding a further string to itself.

h "m2" is created for the last six months of the year.

i And completed by concatenating another string.

j This line embeds the algorithm for determining the first day of the year for the 19th century. For 1978 it is 78 + 11 = 89, divided by 4 = 22, plus 78 = 100, divided by 7 = 14, with a remainder (from another implicit variable "*rmdr") of 2. Then "incr" = 5. The first Monday of the first fiscal week of a year has the ordinal value of (7-incr).

k This line is not executed, because it starts with "if mo is Less Than 7", and "mo" is 8 for August.

l This line is executed. It shows a scan operator in action. ">" means to scan to the right, and "<" means scan to the left. But first the function within the nested parentheses must be evaluated, getting a value of 20 ((8-6)*10). The variable m2 is then truncated right 20 positions, yielding "   July   August". Now all parenthetical evaluation is complete, and "<'" says "scan from the right, to the left, saving the right part when a space is found". This leaves the string "August".

Now we see that declaring the variable "m2"

created a vector of elements, each of variable length!

m "tempord" is declared to have a value derived from using only the rightmost three (212) of the first 24 (8*3) characters of "del", adding the numeric day of month, and adding "incr", which reflects year start.

n From March on (mo Greater Than 2), and if it's a leap year (Leap EQuals yr), "tempord" gets bumped by 1.

o "fw" (fiscal week) is the integer quotient of "tempord" divided by 7. "fd" (fiscal day) is the remainder from the division (*rmdr wasn't used) plus 1.

Finally, the ordinal day of the year is obtained by subtracting "incr". These contortions were caused by fiscal weeks starting on Monday, not Sunday. Because we do it by division, Sunday must occur as the 7th day of the fiscal week, not the 0th.

p The word prefix for the day of the week is picked up. E.g., for Sunday the fiscal day is 7. So we pick up all 42 characters of "dy", save the righthand 6, and scan right until we come to a character that is *not* (^) a space, and save the righthand part, which is "Sun". Then the string "day" is concatenated.

q Now we pick up the correct ordinal suffix from "suf", as defined in line "c". Appending it to the numeric value for the fiscal day, we get the ordinal value, which is reassigned to "fd". Now, however, the value of "fd" goes from numeric to string, and the processor redefines the variable type.

r TEX arithmetic produces answers in normal form only. Results have no leading zeros. Here "ord" is forced to 2 digits so a test can be made on both the tens (X) and the units (I) positions. The string character "0" is concatenated in front, and the two positions extracted. Also, the suffix for the ordinal date is set to "th".

s If the units value is 1, and the tens value is not 1, the suffix is "st" (1st, 21st, ... 91st). For 11 only, it is 11th.

t Similarly, it's 12th, but 2nd, 22nd, etc.

u Similarly, it's 13th, but 3rd, 23rd, etc.

v The suffix is now appended.

The "substitution mode" is put in force, with the vertical bar as the substitution delimiter for this time. From now on, until the mode is turned off with "nosubs", the TEX processor checks every line before execution, to see if it has pairs of this character. If so, the variable name(s) thus delimited has its value substituted before the line is executed. If the variable content thus substituted is executable, it is executed in its turn.

w The delimited (by double quotes in this case) string is printed. If that string has been the content of a named variable, "out:variablename" would do. Four substitutions are made before display.

x After two substitutions, the second display line.

y After one substitution, the third display line.

z "*time" is another read-only variable from the computer system (it has its own time-of-day clock) with the format "hh:mm:ss". "sec" gets its value from the "ss" part, "hm" getting "hh:mm".

aa If the seconds value has not exceeded 25, the "hm" value is good because it requires no rounding (the value is 25, not 30, to approximate system response and printing the first three lines of Figure 4). Here a jump to the label "lend" displays the last line, turns the subs mode off so as to not make succeeding programs act erroneously, and returns.

bb Else "min" is extracted as the "mm" value of "hm", and "hr" extracted as the "hh" part. Rounding up is done by "min + 1", but again an arithmetic operation yields normal form. So possible single digit conditions must be preceded by a "0", and truncated left 2 positions from the right.

cc This could yield a rounded value of 60, so we must go to the next hour, set "min" to "00", and add 1 to the hour.

dd Hour roundup might result in going into the next day. But it wouldn't warrant going back to correct what was already displayed, because that might have been perfectly true when it came out.

ee Now the hour-minute time is reconstructed.

ff The standard lines of spaces preceding labels.

gg After one substitution the last display line is put out. A Line Feed is added for easier display reading.

hh Again a line of spaces before a labeled line.

ii Every TEX program (without exception) has a label "explain". It's especially useful when keywords that describe programs are collected in a file. Then an inventory program can ask for the keywords describing the process you want to do. Getting one or more program names with a lot of hits, you'll want to know more explicitly what they do. So "call program!explain", which starts to execute "program" at the label "!explain". As you can see from the rest of the lines of this program, it is just a series of "out" to display.

That's not the only use. Intermediate labels in the "explain" section can be used to give selected reminders when the called program is used incorrectly.

UTILITY OF TEX

TEX is used for many different types of applications. It's great for "programmerless" computing in large systems where one would otherwise have to learn a Job Control Language. Instead, the needed input is asked for interactively, and the replies are used to tailor a JCL pattern and run the job automatically.

It's been used for relational databases, computer-assisted learning, a software factory, and design prototyping of software. It's easy to teach, easy to use, and one can usually write an application in ⅕ to ⅒ of the time required to use most other languages.

In this brief introduction we haven't yet shown how the editor portion is used. You get two quickies for a start.

The one in Figure 5 will list line number and length for a source program in BASIC, etc. Just a few clues are needed to understand it — "*cl" is the current line, "*lcl" is its length, *eof is the end-of-file condition, and "f;1" means move the pointer forward 1 (go to the next line).

```
count=0

!loop if *eof out:"Done" return
count=count+1 out:count,"  ",("00",*lcl)['2,"  ",*cl
f;1 goto !loop
```

Figure 5

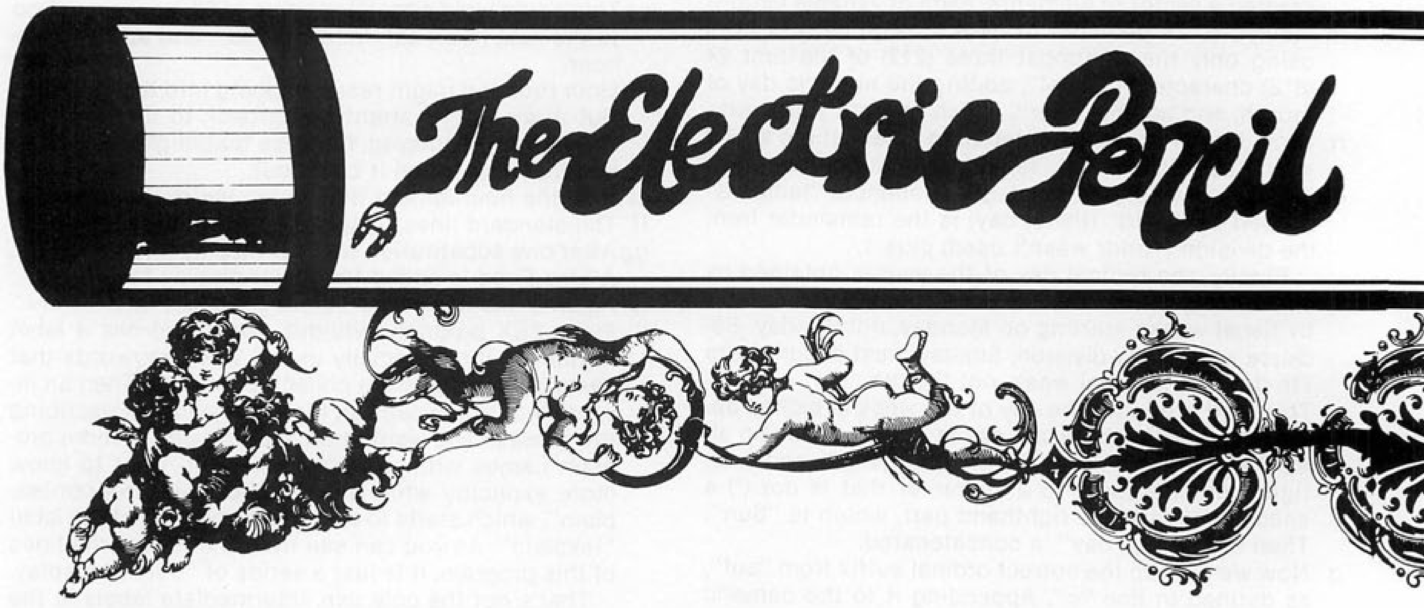
```
!elimdup a=*cl f;1

!all if *eof out:"Task complete." return
b=*cl if a:eqs:b d goto !all
a=b f;1 goto !all
```

Figure 6

The one in Figure 6 will eliminate duplicate lines in an ordered list. The new clues here — "eqs" means "equals the string", and "d" means delete the current line.□

REFERENCES

1. The TEX Subsystem of the Timesharing System, Series 60 Level 66, Honeywell Information Systems, 200 Smith Street, Waltham, MA 02154, Order DF72.



INTRODUCTION

Text editors are typically line-oriented computer programs that can be used to create and alter other computer programs. The smaller BASIC interpreters have editing commands that are limited to the deletion of the previously typed character and the replacement of a complete line. Extended BASIC and some assemblers have full text editors that allow characters to be inserted into or deleted from existing lines. Editors such as these are suitable for line-oriented text typical of BASIC and assembly-language source programs.

Ordinary text, such as this article, is paragraph oriented rather than line oriented. If a few words are added to one line, all of the remainder of the lines in that paragraph may have to be moved over. Furthermore, if the right-hand margin must be aligned (right justified), extra spaces must be inserted between words. Line-oriented editors are not suitable for these tasks.

Michael Shrayer has written a paragraph-oriented text editor that has nearly all the features one could possibly want. Text is continuously typed with no carriage returns or other line indicators. Only the ends of paragraphs and pages are marked. Text is displayed on a video screen as it is typed in. When the text is eventually printed, carriage returns and line feeds are put in the proper place.

NECESSARY HARDWARE

The CP/M version of Electric Pencil can be obtained on an 8-inch, soft-sectored floppy diskette (the standard Digital Research version), or on a 5-inch hard-sectored floppy diskette (Lifeboat Associates version). Several printer/cassette options are available. The following hardware is needed:

1. 8080 or Z80 microprocessor with a minimum of 16K bytes of memory (20K bytes for the Lifeboat version).
2. CP/M operating system.
3. Floppy disk(s).
4. Printer
5. Video display module (direct memory display) addressed to CC00 HEX. A regular video terminal won't work.
6. Cassette recorder (optional).

STARTUP

One of the nice features of CP/M (see INTERFACE AGE, July 1978) is the compatibility of software. Load CP/M in the usual way. Replace the diskette in drive A with the Electric Pencil diskette and place an initialized diskette in drive B (if you have more than one drive). Do a warm start with a Control-C (^C) and give the command PENCIL. CP/M will load PENCIL, clear the video screen and display:

THE ELECTRIC PENCIL II © MICHAEL SHRAYER

Typing any character will clear the video screen and put the cursor in the home position (upper left corner).

ENTERING TEXT

As characters are typed, they are simultaneously entered into the edit buffer and displayed on the video screen. Typing a tab (^I) moves the cursor to the next 8-column field. When you get to the end of a line, keep typing. If the current word won't fit at the end of the line, the entire word jumps to the beginning of the next line. If you make a mistake, type DEL (RUB). The cursor backs up, deleting the previous character.

Continue typing until you reach the end of a paragraph, then type a line feed. A back arrow will appear on the video screen and the cursor will jump to the next line. After the screen is full, the text scrolls up one line for each line typed.

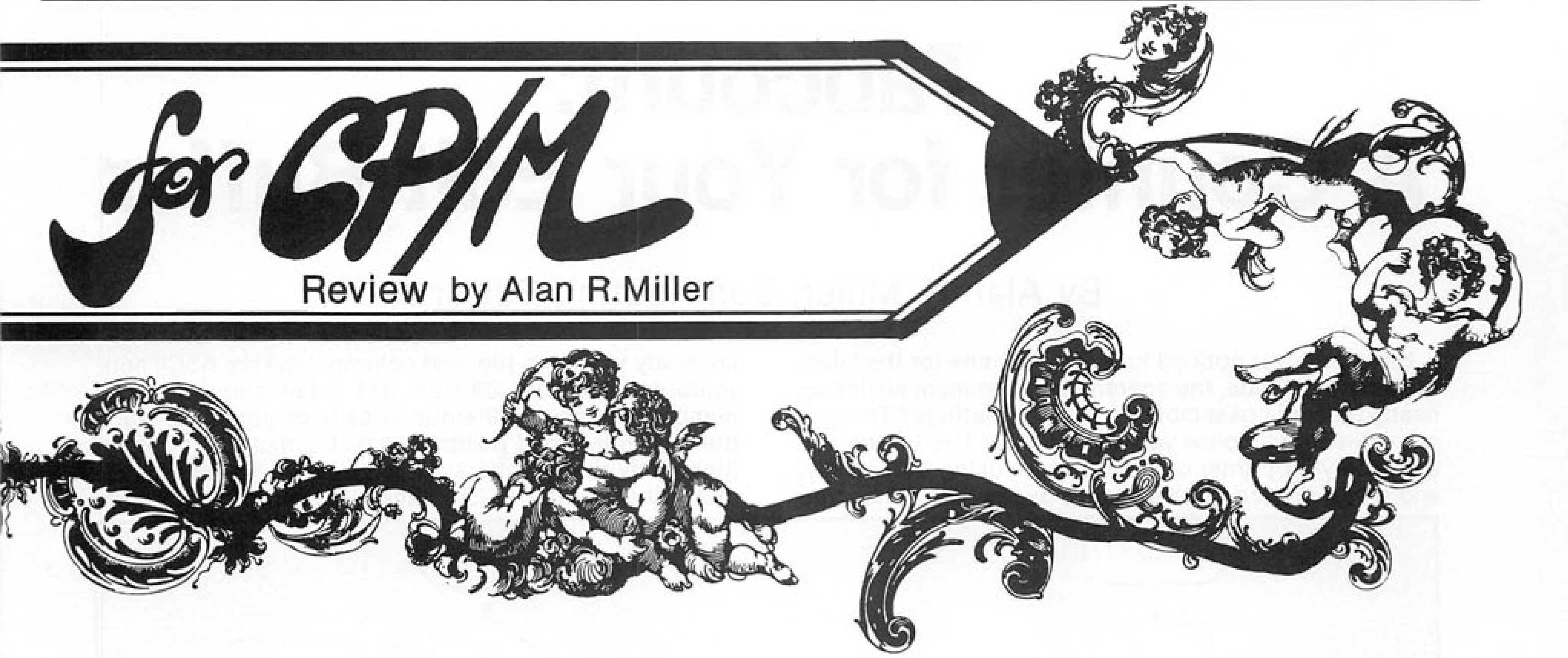
CURSOR-MOVE AND SCROLL COMMANDS

ASCII control characters (see INTERFACE AGE, May 1978) are used to move the cursor on the screen. The cursor on the video screen is moved to the beginning of the line with a carriage return and to the HOME position (the upper left corner) with a ^Q. Up, down, left, and right are obtained with the respective commands ^W, ^Z, ^A, and ^S. Notice that these commands are related to their respective keyboard positions. Q is in the upper left corner of the keyboard, and the other four are arranged in a diamond pattern of up, down, left, and right.

Scroll up and down are obtained with ^E and ^X. A ^B moves the cursor to the beginning of the text, and a ^N moves it to the end. In this latter case, no text is visible unless the cursor is moved up one line with ^W.

SCROLLING CONTROL

The scrolling speed can be changed by typing a number from 1 to 9 (9 is slowest). Pressing the space bar will halt the scrolling, and typing a carriage return will cause scrolling to resume. The scrolling direction can be re-



versed by typing the appropriate command (^E or ^X). Typing a zero will halt the scrolling. Now, each subsequent pressing of the space bar will display the next 16 lines. A carriage return will resume scrolling.

TEXT-ALTERATION COMMANDS

The text-alteration commands operate with respect to the cursor. The character under the cursor is deleted with a ^D, and all the text from this point to the end of the paragraph is moved over. There are two line-delete commands. A ^Y removes the entire line the cursor is on, while a ^T deletes from the cursor to the line end. Characters are inserted by typing a ^F. The insert mode is terminated by typing another ^F or by moving the cursor off the line.

A block of text is defined by placing back-slash markers at each end. This block can then be moved with a ^H or deleted with a ^U.

STRING SEARCH

A ^V clears the screen and displays:

SEARCH STRING?

Type the string and a carriage return. The next occurrence of the string will appear at the top of the screen. The search can be continued by typing ^C. If you want to replace a string with another, type the original string, a regular slash for a separator, and the replacement string. You have to remember not to search for a string containing a slash. If you attempt to search for:

Y/N

PENCIL will interpret the slash as a replacement command and change all occurrences of Y with N. The # symbol can be used as an ambiguous character in the search string.

Commands can be repeated by using ^R. For example, a row of minus signs can be entered with ^R40-.

DISK AND TAPE COMMANDS

Type a ^K and the system enters the disk/tape mode. The video screen displays a list of the commands and the default disk drive number. Commands are now given using printable ASCII characters, not control characters.

D2	List directory for drive 2
S filename	Save edit buffer with given name
K filename	Delete filename (kill)
L filename	Load filename from disk

The S command generates files of type .PCL, and the K and L commands look for files of the same type. Separate portions of text can be combined into a complete file by giving a series of load commands.

Backup files can be created on magnetic tape and subsequently reloaded. There are three massive erase commands: delete from beginning of buffer to cursor, delete from cursor to end, and delete entire file. You can change diskettes without worrying about a warm start and return to the regular edit mode by typing ESC.

PRINT COMMANDS

When the text in the edit buffer is in its final form, type a ^P. PENCIL switches to the print mode, displaying a list of the commands. Remember, the text in the edit buffer runs continuously. There is no indication of the end of a line; only the ends of paragraphs and pages are marked.

The print format is specified by giving a string of commands on a single line. These commands can be used to set the page length, line length, line spacing, and left margin. There is also a command to right justify the text. If you want to abort the printing because the print parameters are incorrect, type ESC.

POSSIBLE PROBLEMS

It takes more than the 32 ASCII control characters to implement all of the many features of ELECTRIC PENCIL. One way to handle this is to use two characters, a control character followed by a printing character. The print and disk commands and the string search and repeat commands are formed in this way. Unfortunately, there are several places where a text character can be misinterpreted as a command. You must be careful in using the ASCII printing characters dollar sign, period, pound sign, backslash, forward slash, vertical, and underline. As already mentioned, the use of the forward slash in a string search can be a disaster. A period following a paragraph end can stop the printing of the text.

SUMMARY

The ELECTRIC PENCIL is a very versatile, paragraph-oriented text editor. The coupling with CP/M makes it especially powerful. Separate files can be saved on disk and concatinated into a finished file. The output formatting is only limited by your imagination.☐

Tabcount: A Counter for Your Edit Buffer

By Alan R. Miller, Contributing Editor

Have you ever noticed how the columns for the label, the operation code, the operand and comment all line up neatly on some assembly and source listings? The program described below will allow you to line up the columns on your printer or video screen using the tab key, and it can be done like a regular typewriter. When you are ready to skip to the next column, type the ASCII horizontal tab Control-I (09 HEX, 011 octal), and the proper number of spaces will automatically be printed to reach the next print-head position that is a multiple of eight. Since only the tab character is stored in the edit buffer rather than all the ASCII blanks, the buffer size will be

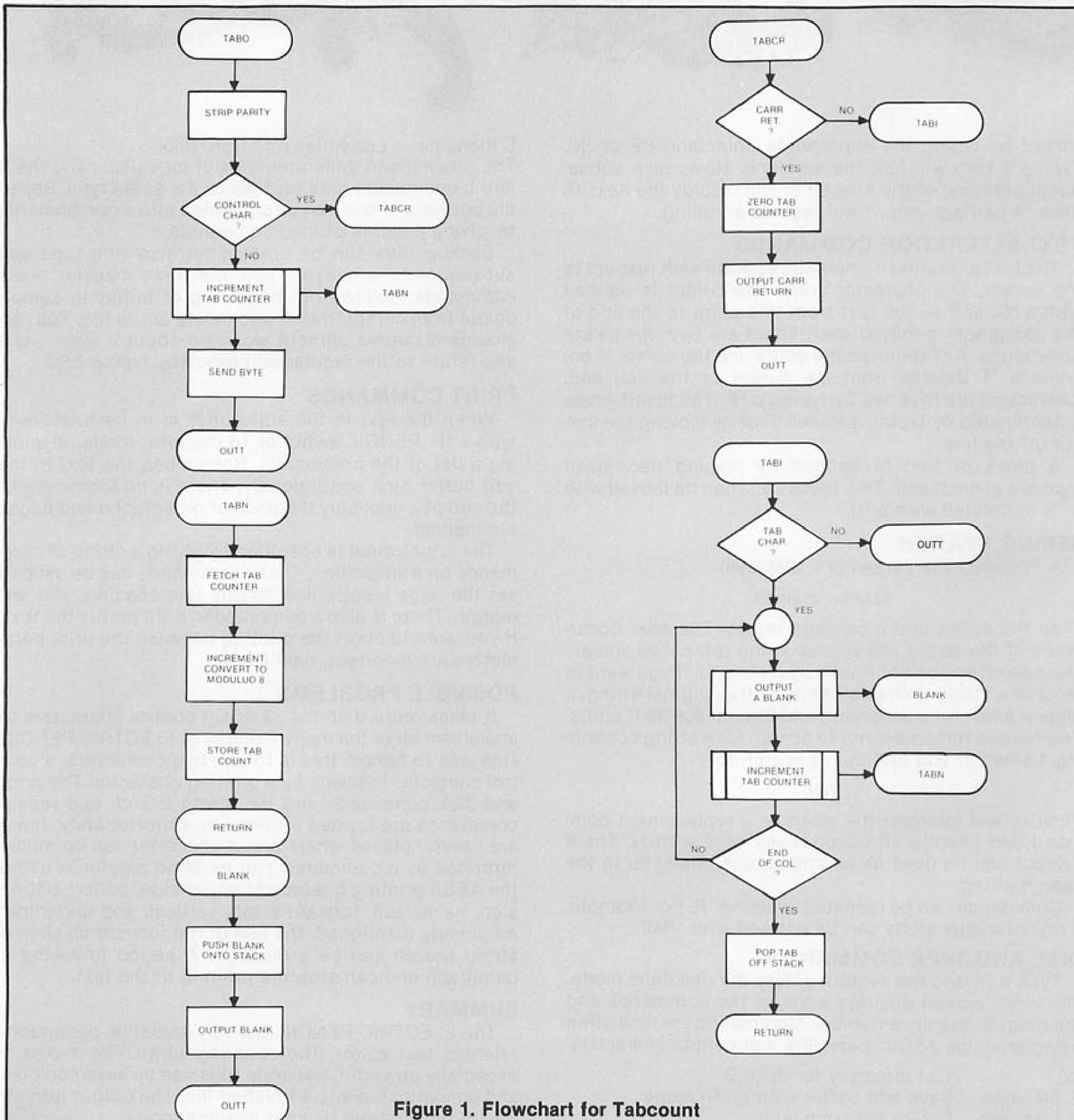


Figure 1. Flowchart for Tabcount

considerably smaller. When the edit buffer is listed, this program will again expand the tabs to print the correct number of blanks.

Tabcount is assembled for location 5C00 and requires 59 bytes of memory. Patch the jump to the output routine so it will jump to location 5C00 with the byte to be outputed pushed onto the stack. (A call would push the stack down two more bytes since the return address would also be pushed onto the stack.) Patch locations 5C0D, 5C23, 5C28 and 5C39 (OUTT) with the address of your regular output routine, or reassemble the program with the EQUATE OUTT set to your output routine.

Tab expansion routine can be placed in ROM or protected memory if the tab-counter byte TABC is placed in read/write memory.

If your output routine expects the outputed byte to be in a register (e.g., the C register) rather than on the stack, define the label OUTT to beat the end of Tabcount:

```
OUTT:  POP   PSW    ;RETRIEVE BYTE

       MOV   C,A    ;PUT BYTE IN C
       JMP   OUT2   ;JUMP TO OUTPUT ROUTINE
```

and transfer the byte from the stack to the required register. Also change the first line:

```
TABO:  MOV   A,C    ;FETCH BYTE
```

to put the byte into the accumulator.

None of the general registers B/C, D/E, or H/L are changed, but two additional levels of stack (4 bytes) must be available in your calling program. Your regular output routine should look like this:

```
OUTT:  IN    STATUS   ;CHECK PRINTER STATUS
       ANI   INMSK    ;MASK FOR TRANSM BUFF EMPTY
       JZ    OUTT     ;LOOP UNTIL READY
       POP   PSW      ;RETRIEVE BYTE FROM STACK
       OUT   DATA     ;PRINT BYTE
       RET            ;DONE
```

Your calling program may have to be modified so that it will accept the tab-control character as input, will put it in the edit buffer, and then will echo it to the printer. At some point after inputing a character, your program will check to see if the byte is a control character such as carriage return, line feed, etc.:

```
.....
       CPI   " "      ;COMPARE TO A BLANK
       JC    CONTR    ;JUMP IF LESS (CONTROL CHAR)
.....
```

The branching here is to prevent the echoing of control characters. This is part of the code that will have to be changed so that the ASCII tab character is actually sent to the output routine:

```
       .....
       CPI   " "      ;COMPARE TO A BLANK
       JC    TAB3     ;JUMP IF CONTROL CHAR
TAB2:  ......         ;NON CONTROL CHARACTER
       .......
TAB3:  CPI   9        ;ASCII TAB?
       JZ    TAB2     ;JUMP IF TAB
       JMP   CONTR    ;OTHER CONTROL CHAR
```

This tab expansion routine can be placed in ROM or protected memory if the tab-counter byte TABC is placed in read/write memory.

PROGRAM LISTING

```
                ;TABCØUNT:  A PRØGRAM TØ EXPAND TABS (CØNTRØL-I)
                ;
                ;PRØGRAMMED FØR AN 8080 MICRØPRØCESSØR
                ;BY ALAN R. MILLER
                ;NEW MEXICØ TECH,  SØCØRRØ,  NM  87801
                ; 505-835-5619       SEPTEMBER 22,1977
                ;
                ;USAGE:
                ;  THIS PRØGRAM IS USED TØ EXPAND THE ASCII TAB
                ;FUNCTIØN.  EDIT BUFFERS CAN BE GREATLY REDUCED
                ;IN SIZE BY STØRING ØNLY THE TAB (CØNTRØL-I),
                ;09 HEX.  CALL THE BEGINNING ØF THIS PRØGRAM,
                ;"TABØ", WITH THE ØUTPUT BYTES PUSHED ØNTØ THE
                ;STACK.  ADDRESS "ØUTT" TØ YØUR ØUTPUT RØUTINE
                ;WHICH SHØULD LØØK LIKE THIS:
                ;
                ;       ØUTT:   IN      STATUS  ;CHECK STATUS
                ;               ANI     ØMSK    ;MASK FØR ØUTPUT
                ;               JZ      ØUTT    ;LØØP UNTIL READY
                ;               PØP     PSW     ;RETRIEVE BYTE
                ;               ØUT     DATA    ;ØUTPUT IT
                ;               RET
                ;
                ;  TYPING A TAB WILL PRINT BLANKS UNTIL THE PRINT
                ;HEAD IS LØCATED A MULTIPLE ØF EIGHT SPACES
                ;FRØM THE LEFT.
                ;
                ;YØU MAY HAVE TØ ALTER YØUR INPUT RØUTINE
                ;TØ ACCEPT A CØNTRØL-I.
                ;
                ;EQUATES
                ;
5C00            ØUTT    EQU     0F80BH  ;REGULAR ØUTPUT RØUTINE
5C00            TAB     EQU     9       ;ASCII HØRIZØNTAL TAB
5C00            CR      EQU     0DH     ;CARRIAGE RETURN
                ;
                ;PATCH ØUTPUT RØUTINE TØ JUMP HERE WITH
                ;ØUTPUT BYTE IN ACCUMULATØR
                ;
5C00 F1         TABØ:   PØP     PSW     ;FETCH BYTE
5C01 F5                 PUSH    PSW     ;SAVE IT
5C02 E67F               ANI     7FH     ;STRIP PARITY
5C04 FE20               CPI     " "     ;SEE IF CØNTRØL CHARACTER
5C06 DA195C             JC      TABCR   ;JUMP IF CØNTRØL CHAR
5C09 CD0F5C             CALL    TABN    ;INCREMENT TAB CØUNTER
5C0C C30BF8             JMP     ØUTT    ;ØUTPUT CHARACTER
                ;
                ;SUBRØUTINE TØ INCREMENT THE TAB CØUNTER
                ;AND REDUCE THE RESULT TØ MØDULØ EIGHT
                ;
5C0F 3A3B5C     TABN:   LDA     TABC    ;FETCH TAB CØUNTER
5C12 3C                 INR     A       ;INCREMENT IT
5C13 E607               ANI     7       ;MØDULØ EIGHT
5C15 323B5C             STA     TABC    ;SAVE IT
5C18 C9                 RET
                ;
                ;CHECK FØR CARRIAGE RETURN
                ;IF SØ, ZERØ THE TAB CØUNTER
                ;
5C19 FE0D       TABCR:  CPI     CR
5C1B C2255C             JNZ     TABI    ;JUMP IF NØT CR
5C1E AF                 XRA     A       ;GET A ZERØ
5C1F 323B5C             STA     TABC    ;ZERØ TAB CØUNTER
5C22 C30BF8             JMP     ØUTT    ;ØUTPUT CR
                ;
                ;CHECK FØR TAB (CØNTRØL-I) CHARACTER
                ;
5C25 FE09       TABI:   CPI     TAB
5C27 C20BF8             JNZ     ØUTT    ;JUMP IF NØT TAB
5C2A CD355C     TAB2:   CALL    BLANK   ;ØUTPUT A BLANK
5C2D CD0F5C             CALL    TABN    ;INCREMENT THE TAB CØUNTER
5C30 C22A5C             JNZ     TAB2    ;LØØP UNTIL END ØF TAB
5C33 F1                 PØP     PSW     ;REMØVE TAB FRØM STACK
5C34 C9                 RET
                ;
                ;SUBRØUTINE TØ ØUTPUT A BLANK
                ;
5C35 3E20       BLANK:  MVI     A," "   ;GET A BLANK
5C37 F5                 PUSH    PSW     ;PUT ØNTØ STACK
5C38 C30BF8             JMP     ØUTT    ;ØUTPUT A BLANK
                ;
5C3B 00         TABC:   DB      0       ;TAB CØUNTER
                ;
                        END     TAB
```

```
2C50 T  A  B  Ø  :  09 P  Ø  :P 09 P  S  W  09 ;  F
2C60 E  T  C  H     B  Y  T  :E 0D 09 P  U  S  H  09
2C70 P  S  W  09 ;  S  A  V  :E    I  T  0D 09 A  N
2C80 I  09 7  F  H  09 ;  S  :T R  I  P     P  A  R
2C90 I  T  Y  0D 09 C  P  I  :09 "     "  09 ;  S  E
2CA0 E     I  F     C  Ø  N  :T R  Ø  L     C  H  A
2CB0 R  A  C  T  E  R  0D 09 :J C  09 T  A  B  C  R
2CC0 09 ;  J  U  M  P     I  :F    C  Ø  N  T  R  Ø
```

Figure 2. ASCII Dump of Part of an Edit Buffer Demonstrating Use of the Tab Functon (09 HEX).

An ASCII dump of the edit buffer for this program is shown in part in Figure 2. (I wrote my ASCII dump so that non-printing characters are represented in hexadecimal.)☐

Date and Time for the CP/M

By W. C. Hoffer

Nearly all computer systems must know the date and the time of day. The software described here provides an interface between the CP/M operating system from Digital Research and the COMPU/TIME board manufactured in Huntington Beach, California. Since CP/M is hardware independent, I will not go into the details of that system.

The code shown in the listing titled SETIME.PRN provides the ability to set the date and time on the board. When the board is purchased, some software routines are provided. I have taken these and added the wants and needs of CP/M along with my own preferences. Particularly important is that the board accepts invalid dates and times and then hangs up with no indication that there is a problem. Instead of writing all of the code required for input validity checks, I have chosen to warn the user and provide a program abort capability.

Some code in this program is duplicated (READ DATE and READ TIME sections). Initially, this program was written for the purpose of lifting sections of code that could "stand alone" to be used in other programs. This method has worked well for me since hardly any change is required after the initial development. The program checks to see if there is a board in the system, and if there is none, it writes eight zeros into the output buffer. When a board is present, the date and time are set and held. The user is then prompted to strike a key for ZERO SECONDS SYNC, allowing the clock to be set accurately using a known time source. The complete setting procedure takes approximately two minutes and forty seconds. This means that you should input a time that is about two minutes later than the current time. If the program has not prompted for ZERO SECONDS SYNC after about two minutes, you should abort the mission by striking any key and check your input. Please note that all of the listings provided are full of comments that can answer many of your questions. An actual setting is shown in Figure 1.

```
A>SETIME

INPUT MONTH, DAY,HOURS,MINUTES
IN THE FORM MM,DD,HH,MM (MUST BE TWO DIGITS EACH)
INVALID INPUT IS NOT CHECKED AND WILL HANGUP THE BOARD
02,27,12,48
STRIKE ANY KEY TO ABORT THIS PROGRAM AND RETURN TO CP/M

STRIKE KEY FOR ZERO SECONDS SYNC

NOW CHECK THE SETTING

DATE 02/27/78 TIME 12:48:00

A>
```

FIGURE 1 "SETIME" EXAMPLE

The program titled TIME.PRN shows the software that displays the date and time on the console device. The appropriate routines were taken from the SETIME program. Figure 2 shows a request and the result.

```
A>TIME

DATE 02/27/78  TIME 13:02:31

A>
```

FIGURE 2 "TIME" EXAMPLE

TIMESUBS.PRN is a listing of the READ DATE and READ TIME software set up as subroutines which I have stored in PROM beginning at E000H. This allows the user to call for the date or time from a running program. The calling program must point at the starting address of an 8-byte buffer with the D and E registers. The routines will return an eight byte ASCII date or time.

Now that the date and time can be set and the results can be displayed on the console, we can look at some applications. First, CP/M should display the date and time during a cold boot and a warm boot. These modifications dictate the need for the source code of BIOS. Since the TIMESUBS software is located outside of BIOS, I set aside two vectors in the ENTRY POINT TABLE. Thus, no changes to applications software will be required if TIMESUBS has to be relocated. I left room for expansion in the ENTRY POINT TABLE before establishing my vectors.

I have the IMSAI supplied Version 1.33 of CP/M, and the modifications I have made work very well. If you do not have the same version, you should have no problem incorporating the changes into your BIOS, assuming you have the source code for BIOS and have some insight as to its relation to the CP/M system.

My vector for DATE is at the beginning of BIOS + 45H, and my vector for TIME is at BIOS + 48H. Both vectors point to the TIMESUBS entry points at E000H and E003H. Figure 3 shows the vectors and all of the code that can be placed anywhere in BIOS. The code in Figure 4 must be placed such that it will be executed each time there is a jump to the WARM boot entry point.

```
BD00  C38DBF      ENTAB:  JMP  COLD      ;COLD START RETURN
BD03  C36EBF              JMP  WBOOT     ;COME HERE FOR REBOOT (VIA 0)
BD06  C3F2BD              JMP  CONSTAT
Bd09  C30ABE              JMP  CONIN
BD0O  C318BE              JMP  CONOUT
BD0F  C326BE              JMP  LIST
BD12  C336BE              JMP  PUNCH
BD15  C347BE              JMP  READER
BD18  C3CBBD              JMP  HOME
BD1B  C3E4BD              JMP  SELDSK
BD1E  C3D1BD              JMP  SETTRK
BD21  C3D6BD              JMP  SETSEC
BD24  C2DBBD              JMP  SETDMA
BD27  C392BD              JMP  READ
BD2A  C39CBD              JMP  WRITE
BD2D  C350BF              JMP  NXM       ;FOR RESTART 7: GIVE ERR MESSAGE
BD30  C395BF              JMP  WARM      ;WARM BOOT RETURN - FINISH INIT
BD33                      DS     18      ;ROOM FOR EXPANSION VECTORS
                  ;
                  ;ANY CHANGES TO THESE VECTORS RELATIVE TO THEIR POSITION
                  ;IN THE'ENTRY POINT TABLE'AFFECT THE LIST PROGRAMS
                  ;AND ANY OTHER PROGRAMS THAT USE THESE VECTORS FOR DATE & TIME
                  ;
BD45  C300        GDATE:  JMP     DATE   ;8 BYTE DATE MM/DD/YY
BD48  C303E0      GTIME:  JMP     TIME   ;8 BYTE TIME HH:MM:SS
                  ;
                  ; SIGN-ON MESSAGE, TYPED AFTER RETURN FROM BOOT
                  ;
BD4B  0D0A34384BMESSAGE:  DB CR,LF,'48K CP/M EXPERIMENTAL VERS 1.33.3  ',0
                  ;
                  ;DATE & TIME MESSAGE
                  ;
BD71  0D0A444154DATEMESS:        DB     CR,LF,'DATE'
BD78  58582F5858DATES            DB     'XX/XX/XX  TIME '
BD87  58583A5858TIMES:           DB     'XX:XX:XX',CR,LF,'$'
```

FIGURE 3 "BIOS" ADDITIONS

```
                  ;
                  COLD:           ;"BOOT" RETURNS HERE AFTER COLD START
BF8D  214BBD              LXI  H,MESSAGE ;COLD START SIGN ON
BF90  CDBABF              CALL CONOMSG
BF93  0E00                MVI  C,0       ;COLD STARTS FROM DRIVE ZERO
                  WARM:           ;"BOOT"RETURNS HERE AFTER WARM START
                  ;
                  ;GET TIME & DATE AND PRINT IT
                  ;
BF95  C5                  PUSH    B            ;PRESERVE IOBYTE & SELECTED DISK
BF96  1178BD              LXI     D,DATES      ;
BF99  CD45BD              CALL    GDATE        ;GET THE DATE
BF9O  1187BD              LXI     D,TIMES      ;
BF9F  CD48BD              CALL    GTIME        ;GET THE TIME
BFA2  1171BD              LXI     D,DATEMESS   ;
BFA5  0E09                MVI     C,9          ;
BFA7  CD0500              CALL    ENTRY        ;PRINT IT
BFAA  C1                  POP     B            ;
```

FIGURE 4 "BIOS" MODIFICATIONS

Disk Operating System

When these modifications have been incorporated into the system, the date and time will be displayed on the console device each time either a cold boot or warm boot is initiated. Anytime you want the date and time you need only to warm boot (control C). This means that you no longer need the TIME program described earlier. Figure 5 is an example of a cold boot and a warm boot.

Everyone involved in software development is plagued with keeping track of the latest listing of the program under development. Many times, I have thrown away the wrong listing and ended up having to get a new listing just to be sure. Having the date and time along with the name of the program at the top of each page of a listing will save a good deal of time.

```
48K CP/M EXPERIMENTAL VERS 1.33.3
DATE 02/27/78  TIME 13:08:56

A>

          "COLD BOOT"

^C

DATE 02/27/78  TIME 13:09:10

A>

          "WARM BOOT"
```

FIGURE 5 CP/M COLD BOOT AND WARM BOOT EXAMPLES

Incorporating the date and time into a LIST program is also a good application for my new software. The LIST program I use was provided by IMSAI and since it bears a copyright message, I won't supply the listing. If you have Version 1.3 of LIST, you can incorporate these changes directly. If you do not have this Version, you can apply the techniques and some of the code. The code in Figure 6 is self-explanatory and can be placed anywhere in the program. Figure 7 shows TITLBUF in its original form, and Figure 8 shows the changes to make and the addition of the storage area DATEBUF. Figure 9 shows the calls to DATE and TIME which are placed at CIB9 and the two statements that must be added to the PAGE NUMBER routine. The code in Figure 10 can be inserted anyplace where it won't be executed since it is used as a subroutine. This code determines the size of the CP/M system that is currently running and then jumps to the proper place in the ENTRY POINT TABLE of BIOS. This method can be used anytime a program uses the ENTRY POINT TABLE directly. The program listings included in this article were prepared using my list program with date and time.

```
            ;
            ;LOCATIONS 1 & 2  CONTAIN THE ADDRESS OF THE WARM START VECTOR
            ;
0001 =      LOC1    EQU     1       ;LOCATION CONTAINING THE ADDRESS OF BIOS+3
            ;
            ;OFFSET IN BIOS FOR DATE & TIME VECTORS
            ;BECAUSE WE CAN ONLY DETERMINE THE START OF BIOS+3
            ;THE BELOW MUST BE ADJUSTED BY 3
            ;
0042 =      DADDR   EQU     42H     ;ACTUALLY IT'S 3 MORE
0045 =      TADDR   EQU     45H     ;DITTO
```

FIGURE 6 "LIST" ADDITIONS

```
0386 00          TITLBUF: DB   0          ;SAYS NO TITLE HERE YET
0387                      DS   100H                ;REST OF TITLE BUFFFER
```

FIGURE 7 ORIGINAL "TITLBUF"

```
                 TITLBUF:
03D9                      DS   100H
                                          ;TITLE BUFFFER
                 DATEBUF:
04D9  0D0A204441          DB
                            CRC,LFC,' DATE XX/XX/XX  TIME XX:XX:XX',CRC,LFC,0
```

FIGURE 8 CHANGES TO "TITLBUF" AND ADDITION OF "DATEBUF"

```
                 C1B9:
                 ;
                 ;GET TIME & DATE
                 ;
01A0  11E104             LXI     D,DATEBUF+8
01A3  CD3E01                     CALL    DATE
01A6  11F004             LXI     D,DATEBUF+23
01A9  CD4801                     CALL    TIME



                 ;PAGE NUMBER
033A  3A3B01             LDA PAGNUMELAG
033D  B7                 ORA A
033E  CA6003             JZ NOPAGNUM
                         ;SPACE TO COLUMN
0341  3E20       PAGN1:  MVI A,' '
0343  CD8C02             CALL LSTCH                 ;MINIMUM ONE SPACE
0346  3AF504             LDA COL
0349  DF41               SBI PAGNUMCOL
034B  FA4103             JM PAGN1
                         ;"PAGE" TEXT
034E  217C03             LXI H,PAGETXT
0351  CD8203             CALL LSTRING
                         ;NUMBER
0354  2AF704             LHLD PAGE
0357  CD8E03             CALL DECPR
                 ;
                 ;THE NEXT TWO STATEMENTS WERE ADDED TO GET DATE & TIME
                 ;AT THE TOP OF THE PAGE
035A  21D004             LXI     H,DATEBUF
035D  CD8203             CALL    LSTRING
```

FIGURE 9 "LIST" ADDITIONS

```
                 ;
                 ;DATE & TIME ROUTINES-DETERMINES WHERE BIOS IS LOCATED
                 ;AND VECTORS APPROPRIATELY
                 ;
013E  D5         DATE:   PUSH    D       ;
013F  114200             LXI     D,DADDR;
0142  2A0100             LHLD    LOC1    ;
0145  19                 DAD     D       ;
0146  D1                 POP     D       ;
0147  E9                 PCHL            ;JUMP TO BIOS+DADDR
0148  D5         TIME:   PUSH    D       ;
0149  114500             LXI     D,TADDR;
014C  2A0100             LHLD    LOC1    ;
014F  19                 DAD     D       ;
0150  D1                 POP     D       ;
0150  F9                 PCHL            ;JUMP TO BIOS+TADDR
```

FIGURE 10 "LIST" DATE AND TIME SUBROUTINES

PROGRAM LISTING 1

```
                ;       THIS PROGRAM WILL SET THE TIME AND DATE ON THE COMPU/TIME
                ;       BOARD MANUFACTURED AT 8532 HAMILTON AVE.,
                ;       HUNTINGTON BEACH,CA 92646,(714)536-9967
                ;
                ;       THE BOARD MUST BE ADDRESSED BEGINNING AT 'C0' OR YOU MUST
                ;       CHANGE THE CONTROL & DATA PORT ASSIGNMENTS BELOW
                ;
                ;       THIS SOFTWARE EXECUTES ON THE CP/M OPERATING SYSTEM
                ;
                ;       WRITTEN BY W.C.HOFFER-2721 N. WANDA-SIMI VALLEY,CA.-93065
0100                    ORG     100H
                ;
                ;SET UP THE STACK
                ;
0100 210000             LXI     H,0
0103 39                 DAD     SP
0104 221504             SHLD    OLDSP
0107 311705             LXI     SP,STACK
                ;
                ;ASSIGNMENTS
                ;
0005 =          ENTRY:  EQU     5       ;CP/M ENTRY TO BDOS
000D =          CR:     EQU     0DH     ;CARRIAGE RETURN
000A =          LF:     EQU     0AH     ;LINE FEED
00C5 =          ACONT:  EQU     0C5H    ;PORT A CONTROL
00C7 =          BCONT:  EQU     0C7H    ;PORT B CONTROL
00C4 =          ADATA:  EQU     0C4H    ;PORT A DATA
00C6 =          BDATA:  EQU     0C6H    ;PORT B DATA
                ;
                ;CHECK FOR BOARD PRESENT
                ;
010A DBC4               IN      ADATA   ;READ PORT A DATA
010C FEFF               CPI     0FFH    ;BOARD PRESENT?
010E C21601             JNZ     BEGIN   ;YES
0111 2A1504     RETCPM: LHLD    OLDSP   ;NO-RETURN TO CP/M
0114 F9                 SPHL
0115 C9                 RET
                ;
                ;SET DATE & TIME
                ;
                BEGIN:
0116 CDA602             CALL    CLERES  ;RESET CLOCK
0119 CDAD02             CALL    CLEINT  ;INIT THE CLOCK
011C CD6601             CALL    GETINP  ;GET THE INPUT
011F 210A04             LXI     H,MONI  ;LOAD B&C WITH MONTH
0122 46                 MOV     B,M     ;
0123 23                 INX     H       ;
0124 4E                 MOV     C,M     ;
0125 3E77               MVI     A,77H   ;
0127 CD6A03             CALL    SETDIG  ;SET MONTH
012A 23                 INX     H       ;SKIP OVER COMMA
012B 23                 INX     H       ;
012C 46                 MOV     B,M     ;
012D 23                 INX     H       ;
012E 4E                 MOV     C,M     ;
012F 3E57               MVI     A,57H   ;
0131 CD6A03             CALL    SETDIG  ;SET DAY
0134 211004             LXI     H,HOURI ;LOAD B&C WITH HOUR
0137 46                 MOV     B,M     ;
0138 23                 INX     H       ;
0139 4E                 MOV     C,M     ;
013A 3E76               MVI     A,76H   ;
013C CD6A03             CALL    SETDIG  ;
013F 23                 INX     H       ;SKIP OVER COMMA
0140 23                 INX     H       ;
0141 46                 MOV     B,M     ;
0142 23                 INX     H       ;
0143 4E                 MOV     C,M     ;
0144 3E56               MVI     A,56H   ;
0146 CD6A03             CALL    SETDIG  ;SET MINUTES
0149 CD6F02             CALL    SETSEC  ;SYNC ZERO SEC
                ;
                ;DISPLAY DATE & TIME FOR SET CHECK
                ;
014C 116B01             LXI     D,SETMES
014F 0E09               MVI     C,9
0151 CD0500             CALL    ENTRY
                DISPLAY:
0154 21EB03             LXI     H,MON
0157 CD0303             CALL    DATE    ;GET IT
015A 21F103             LXI     H,HOUR
015D CD2503             CALL    TIME
0160 11E703             LXI     D,PDATE ;DISPLAY THE WHOLE THING
0163 0E09               MVI     C,9
0165 CD0500             CALL    ENTRY
0168 C31101             JMP     RETCPM  ;RETURN TO CPM
016B 0D0A4E4F57SETMES:  DB      CR,LF,'NOW CHECK THE SETTING ',CR,LF,'$'
                ;
                ;CONSOLE INPUT ROUTINES
                ;
0166 119F01     GETINP: LXI     D,DMESS ;GET INPUT USING CP/M BUFFERED I/O
0189 0E09               MVI     C,9     ;
018B CD0500             CALL    ENTRY   ;ASK FOR INPUT
018E 110804             LXI     D,TBUFF ;GET THE INPUT
0191 0E0A               MVI     C,10
0193 CD0500             CALL    ENTRY
0196 113202             LXI     D,DMESS1        ;TELL'EM HOW TO ABORT
0199 0E09               MVI     C,9             ;
019B CD0500             CALL    ENTRY           ;
019E C9                 RET
019F 0D0A494E50DMESS:   DB      CR,LF,'INPUT MONTH,DAY,HOURS,MINUTES  ',CR,LF
01C2 494E20544F         DB      'IN THE FORM MM,DD,HH,MM (MUST BE TWO DIGITS EACH)  ',CR,LF
01F7 494E56414C         DB      'INVALID INPUT IS NOT CHECKED AND WILL HANGUP THE BOARD '
                                  CR,LF,'$'
0232 0D0A535452DMESS1:  DB      CR,LF,'STRIKE ANY KEY TO ABORT THIS PROGRAM AND RETURN TO
                                  CP/M ',CR,LF,'$'
                ;
                ;SET SECONDS SYNCHRONIZATION ROUTINE
                ;
026F 118102     SETSEC: LXI     D,SMESS
0272 0E09               MVI     C,9
0274 CD0500             CALL    ENTRY   ;SYNC MESSAGE
0277 0E0B       KWAIT:  MVI     C,11    ;WAIT FOR ANY KEY
0279 CD0500             CALL    ENTRY
027C 1F                 RAR             ;
027D D27702             JNC     KWAIT   ;NOTHING WAITING
0280 C9                 RET             ;SET TIME
0281 0D0A535452SMESS:   DB      CR,LF,'STRIKE KEY FOR ZERO SECONDS SYNC',CR,LF,'$'
                ;
                ;RESET CLOCK/CALENDER PORT
                ;
02A6 3E00       CLERES: MVI     A,0     ;
02A8 D3C5               OUT     ACONT   ;RESET PORT A
02AA D3C7               OUT     BCONT   ;RESET PORT B
02AC C9                 RET
                ;
                ;INITIALIZE CLOCK/CALENDER PORT
                ;
02AD 3E70       CLKINT: MVI     A,70H   ;
02AF D3C4               OUT     ADATA   ;STORE 70H AT PORT A DATA REGISTER
02B1 3E77               MVI     A,77H
02B3 D3C6               OUT     BDATA   ;STORE A 77H AT PORT B DATA REGISTER
02B5 3E14               MVI     A,14H   ;INTERRUPT CODE
02B7 D3C5               OUT     ACONT   ;
02B9 3E04               MVI     A,04H   ;INTERRUPT CODE
02BB D3C7               OUT     BCONT   ;
02BD C9                 RET
                ;
                ;1 HZ WAIT ROUTINE
                ;
02BE DBC6       HWAIT:  IN      BDATA   ;RESET INTERRUPT
02C0 DBC7       WAIT:   IN      BCONT   ;
02C2 E680               ANI     80H     ;CHECK FOR 1 HZ INTERRUPT
02C4 C0                 RNZ             ;RETURN IF YES
02C5 C3C002             JMP     WAIT    ;LOOP IF NOT
                ;
                ;READ A DIGIT ROUTINE
                ;
                RDIGIT:
02C8 7A                 MOV     A,D     ;SELECT DIGIT
02C9 D3C4               OUT     ADATA   ;
02CB DBC4               IN      ADATA   ;RESET INTERRUPT
02CD DBC5       DWAIT:  IN      ACONT   ;TEST FOR DIGIT PRESENT
02CF E680               ANI     80H     ;ANYTHING THERE?
02D1 CACD02             JZ      DWAIT   ;LOOP UNTIL INTERRUPT
02D4 DBC4               IN      ADATA   ;READ A DIGIT
02D6 E60F               ANI     0FH     ;MASK ZONE
02D8 F630               ORI     30H     ;SET ASCII
02DA C9                 RET
                ;
                ;READ FOUR DIGITS ROUTINE
                ;
02DB 1600       READ4:  MVI     D,0     ;SET TO SELECT FIRST DIGIT
02DD CDC802     NEXT:   CALL    RDIGIT  ;DELAY ONE DIGIT SCAN
02E0 CD3F03             CALL    RSDIG   ;READ & STORE DIGIT
02E3 7A                 MOV     A,D
02E4 FE20               CPI     20H     ;TEST IF 2 DIGITS DONE
02E6 C2F902             JNZ     SKIP    ;SKIP A PLACE
02E9 79                 MOV     A,C     ;SEE IF TIME OR DATE
02EA FE00               CPI     0       ;DATE
02EC C2F402             JNZ     COLON   ;IT'S FOR TIME
02EF 3E2F               MVI     A,'/'   ;IT'S FOR DATE
02F1 C3F602             JMP     DOIT    ;
02F4 3E3A       COLON:  MVI     A,':'   ;
02F6 CD0003     DOIT:   CALL    SDIGIT  ;
02F9 7A         SKIP:   MOV     A,D     ;TEST FOR ALL DIGITS DONE
02FA FE40               CPI     40H     ;
02FC C8                 RZ              ;ALL 4 DONE
02FD C3DD02             JMP     NEXT    ;GET ANOTHER DIGIT
                ;
                ;STORE A DIGIT ROUTINE
                ;
0300 77         SDIGIT: MOV     M,A     ;STORE A DIGIT
0301 23                 INX     H       ;INCR H&L
0302 C9                 RET             ;
                ;
                ;READ DATE ROUTINE
                ;
0303 CD6503     DATE:   CALL    BOARD   ;IS THERE A BOARD
0306 CA4A03             JZ      NOBOARD ;NOPE
0309 CDAD02             CALL    CLEINT  ;
030C 3E00               MVI     A,0     ;SET DATE DISPLAY MODE
030E D3C6               OUT     BDATA   ;
0310 0E00               MVI     C,0     ;TELL READ4 THIS IS DATE
0312 CDDB02             CALL    READ4   ;GET 4 DIGITS
0315 3E2F               MVI     A,'/'   ;
0317 CD0003             CALL    SDIGIT  ;
031A 3E37               MVI     A,'7'   ;SET TENS OF YEARS
031C CD0003             CALL    SDIGIT  ;
031F 3E38               MVI     A,'8'   ;SET UNITS OF YEARS
0321 CD0003             CALL    SDIGIT  ;
0324 C9                 RET
                ;
                ;READ THE TIME
                ;
0325 CD6503     TIME:   CALL    BOARD   ;CHECK FOR BOARD PRESENT
0328 CA4A03             JZ      NOBOARD ;NOPE
032B CDAD02             CALL    CLEINT  ;INITIALIZE THE BOARD
032E 3E40               MVI     A,40H   ;SET TIME DISPLAY MODE
0330 D3C6               OUT     BDATA
0332 0E01               MVI     C,1     ;TELL READ4 THIS IS TIME
0334 CDDB02             CALL    READ4   ;GET 4 DIGITS
0337 3E3A               MVI     A,':'   ;
0339 CD0003             CALL    SDIGIT  ;STORE A DIGIT
033C CD3F03             CALL    RSDIG   ;READ & STORE A DIGIT
                ;
                ;READ AND STORE A DIGIT
                ;
033F CDC802     RSDIG:  CALL    RDIGIT  ;
0342 CD0003             CALL    SDIGIT  ;
0345 7A                 MOV     A,D     ;
0346 C610               ADI     10H     ;STEP TO NEXT DIGIT
0348 57                 MOV     D,A     ;
0349 C9                 RET             ;
                ;
                ;NO BOARD IN THE SYSTEM
                ;
                NOBOARD:
034A 3E30               MVI     A,'0'   ;STUFF 8 ZEROS
034C CD0003             CALL    SDIGIT  ;
034F CD0003             CALL    SDIGIT  ;
0352 CD0003             CALL    SDIGIT  ;
0355 CD0003             CALL    SDIGIT  ;
0358 CD0003             CALL    SDIGIT  ;
035B CD0003             CALL    SDIGIT  ;
035E CD0003             CALL    SDIGIT  ;
0361 CD0003             CALL    SDIGIT  ;
0364 C9                 RET             ;
                ;
                ;SEE IF BOARD PRESENT
                ;
                BOARD:
0365 DBC4               IN      ADATA   ;CHECK FOR BOARD PRESENT
0367 FEFF               CPI     0FFH    ;
0369 C9                 RET             ;
                ;
                ;DIGIT SET ROUTINE
                ;
                SETDIG:
036A D3C6               OUT     BDATA   ;CLOCK SET MODE
036C 5F                 MOV     E,A     ;
036D E620       TEST:   ANI     20H     ;TEST BIT 5(1=HRS/MON;0=MIN/DAYS)
036F CA7703             JZ      MINDAY  ;
0372 1600               MVI     D,0     ;SET D TO ZERO IF BIT 5=1
0374 C37903             JMP     HCALL   ;
0377 1620       MINDAY: MVI     D,20H   ;SET D TO 20H IF BIT 5=0
0379 CDBE02     HCALL:  CALL    HWAIT   ;
037C CD9A03             CALL    CHKABORT        ;CHECK FOR ABORT WANTED
037F CDC802             CALL    RDIGIT  ;READ A DIGIT
0382 B8                 CMP     B       ;CHECK "TENS" DIGIT
0383 C27903             JNZ     HCALL   ;LOOP ON NO MATCH
0386 3E10               MVI     A,10H   ;
0388 82                 ADD     D       ;STEP TO UNITS DIGIT
0389 57                 MOV     D,A     ;
038A CD9A03             CALL    CHKABORT
038D CDC802             CALL    RDIGIT  ;
0390 B9                 CMP     C       ;CHECK "UNITS" DIGIT
0391 7B                 MOV     A,E     ;
0392 C26D03             JNZ     TEST    ;
0395 3E46               MVI     A,46H   ;SET TIME HOLD MODE
0397 D3C6               OUT     BDATA   ;
0399 C9                 RET             ;
                ;
                ;CHECK FOR ABORT WANTED
                ;
                CHKABORT:
039A C5                 PUSH    B       ;SAVE THE REGISTERS
039B E5                 PUSH    H       ;
039C D5                 PUSH    D       ;
039D 0E0B               MVI     C,11    ;SEE IF A KEY HAS BEEN DEPRESSED ON CONSOLE
039F CD0500             CALL    ENTRY   ;
03A2 1F                 RAR             ;
03A3 D2B103             JNC     GOAHEAD ;NO ABORT WANTED
03A6 11B503             LXI     D,ABMESS        ;
03A9 0E09               MVI     C,9             ;
03AB CD0500             CALL    ENTRY           ;PRINT ABORT MESSAGE
03AE C35401             JMP     DISPLAY ;DISPLAY THE ABORTED MESS
                GOAHEAD:
03B1 D1                 POP     D
03B2 E1                 POP     H       ;
03B3 C1                 POP     B       ;
03B4 C9                 RET
03B5 0D0A50524FABMESS:  DB      CR,LF,'PROGRAM ABORTED DURING SET-RESULTS UNCERTAIN ',CR,LF,'$'
                ;
                ;STORAGE AREA
                ;
```

```
03E7 0D0A444154PDATE    DB      CR,LF,'DATE
03EE 5858       MON     DB      'XX'    ;MONTH
03F0                    DS      1       ;
03F1 5858       DAY     DB      'XX'    ;DAY
03F3                    DS      3       ;
03F6 202054494DPTIME    DB      '  TIME '
03FD 5858       HOUR    DB      'XX'    ;HOURS
03FF                    DS      1       ;COLON
0400 5858       MIN     DB      'XX'    ;MINUTES
0402                    DS      1       ;COLON
0403 5858       SEC     DB      'XX'    ;SECONDS
0405 0D0A24             DB      CR,LF,'$'
0408 0D00       TBUFF   DB      13,0
040A            MONI    DS      2
040C                    DS      1
040D                    DS      2
040F                    DS      1
0410            HOURI   DS      2
0412                    DS      1
0413                    DS      2
0415            OLDSP   DS      2
0417                    DS      100H
0517 00         STACK:  DB      0
```

PROGRAM LISTING 2

```
                ;       THIS PROGRAM WILL READ & DISPLAY THE TIME AND DATE FROM A
                ;       COMPU/TIME BOARD MANUFACTURED AT 8532 HAMILTON AVE.,
                ;       HUNTINGTON BEACH,CA.,92646,(714)536-9967
                ;
                ;       THE BOARD MUST BE ADDRESSED BEGINNING AT 'C0' OR YOU
                ;       MUST CHANGE THE PORT ASSIGNMENTS BELOW
                ;
                ;       THIS SOFTWARE EXECUTES ON THE CP/M OPERATING SYSTEM
                ;
                ;       WRITTEN BY W.C.HOFFER-2721 N. WANDA-SIMI VALLEY,CA.-93065
                ;
0100                    ORG     100H
                ;
                ;SET UP THE STACK
                ;
0100 210000             LXI     H,0     ;
0103 39                 DAD     SP
0104 220602             SHLD    OLDSP   ;
0107 310803             LXI     SP,STACK
                ;
                ;ASSIGNMENTS
                ;
0005 =          ENTRY:  EQU     5
000D =          CR:     EQU     0DH
000A =          LF:     EQU     0AH
00C5 =          ACONT:  EQU     0C5H    ;PORT A CONTROL
00C7 =          BCONT:  EQU     0C7H    ;PORT B CONTROL
00C4 =          ADATA:  EQU     0C4H    ;PORT A DATA
00C6 =          BDATA:  EQU     0C6H    ;PORT B DATA
                ;
                ;DISPLAY DATE & TIME
                ;
010A 11DF01             LXI     D,MON
010D CD5E01             CALL    DATE    ;GET IT
0110 11EE01             LXI     D,HOUR
0113 CD8101             CALL    TIME
0116 11D801             LXI     D,PDATE ;DISPLAY THE WHOLE THING
0119 0E09               MVI     C,9
011B CD0500             CALL    ENTRY
                ;
                ;RETURN TO CMP THRU CCP
                ;
011E 2A0602             LHLD    OLDSP
0121 F9                 SPHL
0122 C9                 RET
                ;
                ;READ A DIGIT ROUTINE
                ;
0123 7A         RDIGIT: MOV     A,D     ;SELECT DIGIT
0124 D3C4               OUT     ADATA   ;
0126 DBC4               IN      ADATA   ;RESET INTERRUPT
0128 DBC5       DWAIT:  IN      ACONT   ;TEST FOR DIGIT PRESENT
012A E680               ANI     80H     ;ANYTHING THERE?
012C CA2801             JZ      DWAIT   ;LOOP UNTIL INTERRUPT
012F DBC4               IN      ADATA   ;READ A DIGIT
0131 E60F               ANI     0FH     ;MASK ZONE
0133 F630               ORI     30H     ;SET ASCII
0135 C9                 RET
                ;
                ;READ FOUR DIGITS ROUTINE
                ;
0136 1600       READ4:  MVI     D,0     ;SET TO SELECT FIRST DIGIT
0138 CD2301     NEXT:   CALL    RDIGIT  ;DELAY ONE DIGIT SCAN
013B CD9C01             CALL    RSDIG   ;READ & STORE DIGIT
013E 7A                 MOV     A,D
013F FE20               CPI     20H     ;TEST IF 2 DIGITS DONE
0141 C25401             JNZ     SKIP    ;SKIP A PLACE
0144 79                 MOV     A,C     ;SEE IF TIME OR DATE
0145 FE00               CPI     0       ;DATE
0147 C24F01             JNZ     COLON   ;IT'S FOR TIME
014A 3E2F               MVI     A,'/'   ;
014C C35101             JMP     DOIT    ;
014F 3E3A       COLON:  MVI     A,':'   ;
0151 CD5B01     DOIT:   CALL    SDIGIT  ;
0154 7A         SKIP:   MOV     A,D     ;TEST FOR ALL DIGITS DONE
0155 FE40               CPI     40H     ;
0157 C8                 RZ              ;ALL 4 DONE
0158 C33801             JMP     NEXT    ;GET ANOTHER DIGIT
                ;
                ;STORE A DIGIT ROUTINE
                ;
015B 77         SDIGIT: MOV     M,A     ;STORE A DIGIT
015C 23                 INX     H       ;INCR H&L
015D C9                 RET             ;
                ;
                ;READ DATE ROUTINE
                ;
                DATE:
015E CDC201             CALL    BOARD   ;IS THERE A BOARD
0161 CAA701             JZ      NOBOARD ;NOPE
0164 CDC701             CALL    CLKINT  ;INIT THE BOARD
0167 EB                 XCHG            ;PUT DESTINATION OF DATE IN H&L
0168 3E00               MVI     A,0     ;SET DATE DISPLAY MODE
016A D3C6               OUT     BDATA   ;
016C 0E00               MVI     C,0     ;TELL READ4 THIS IS DATE
016E CD3601             CALL    READ4   ;GET 4 DIGITS
0171 3E2F               MVI     A,'/'   ;
0173 CD5B01             CALL    SDIGIT  ;
0176 3E37               MVI     A,'7'   ;SET TENS OF YEARS
0178 CD5B01             CALL    SDIGIT  ;
017B 3E38               MVI     A,'8'   ;SET UNITS OF YEARS
017D CD5B01             CALL    SDIGIT  ;
0180 C9                 RET
                ;READ TIME & READ AND STORE DIGIT ROUTINES
                ;
0181 CDC201     TIME:   CALL    BOARD   ;CHECK FOR BOARD PRESENT
0184 CAA701             JZ      NOBOARD ;NOPE
0187 CDC701             CALL    CLKINT  ;INIT THE BOARD
018A EB                 XCHG            ;PUT DESTINATION OF TIME IN H&L
018B 3E40               MVI     A,40H   ;SET TIME DISPLAY MODE
018D D3C6               OUT     BDATA
018F 0E01               MVI     C,1     ;TELL READ4 THIS IS TIME
0191 CD3601             CALL    READ4   ;GET 4 DIGITS
0194 3E3A               MVI     A,':'   ;
0196 CD5B01             CALL    SDIGIT  ;
0199 CD9C01             CALL    RSDIG   ;
019C CD2301     RSDIG:  CALL    RDIGIT  ;
019F CD5B01             CALL    SDIGIT  ;
01A2 7A                 MOV     A,D     ;
01A3 C610               ADI     10H     ;
01A5 57                 MOV     D,A     ;
01A6 C9                 RET             ;
                ;
                ;NO BOARD IN THE SYSTEM
                ;
                NOBOARD:
01A7 3E30               MVI     A,'0'   ;STUFF 8 ZEROS
01A9 CD5B01             CALL    SDIGIT  ;
01AC CD5B01             CALL    SDIGIT  ;
01AF CD5B01             CALL    SDIGIT  ;
01B2 CD5B01             CALL    SDIGIT  ;
01B5 CD5B01             CALL    SDIGIT  ;
01B8 CD5B01             CALL    SDIGIT  ;
01BB CD5B01             CALL    SDIGIT  ;
01BE CD5B01             CALL    SDIGIT  ;
01C1 C9                 RET             ;
                ;
                ;SEE IF BOARD PRESENT
                ;
                BOARD:
01C2 DBC4               IN      ADATA   ;
01C4 FEFF               CPI     0FFH    ;
01C6 C9                 RET             ;
01C7 3E70       CLKINT: MVI     A,70H   ;
01C9 D3C4               OUT     ADATA   ;
01CB 3E77               MVI     A,77H   ;
01CD D3C6               OUT     BDATA   ;
01CF 3E14               MVI     A,14H   ;
01D1 D3C5               OUT     ACONT   ;
01D3 3E04               MVI     A,04H   ;
01D5 D3C7               OUT     BCONT   ;
01D7 C9                 RET             ;
                ;
                ;STORAGE AREA
                ;
01D8 0D0A444154PDATE    DB      CR,LF,'DATE
01DF 5858       MON     DB      'XX'    ;MONTH
01E1                    DS      1       ;
01E2 5858       DAY     DB      'XX'    ;DAY
01E4                    DS      3       ;
01E7 202054494DPTIME    DB      '  TIME '
01EE 5858       HOUR    DB      'XX'    ;HOURS
01F0                    DS      1       ;COLON
01F1 5858       MIN     DB      'XX'    ;MINUTES
01F3                    DS      1       ;COLON
01F4 5858       SEC     DB      'XX'    ;SECONDS
01F6 0D0A24             DB      CR,LF,'$'
01F9 0D00       TBUFF   DB      13,0
01FB            MONI    DS      2
01FD                    DS      1
01FE                    DS      2
0200                    DS      1
0201            HOURI   DS      2
0203                    DS      1
0204                    DS      2
0206            OLDSP   DS      2
0208                    DS      100H
0308 00         STACK:  DB      0
```

PROGRAM LISTING 3

```
                ;       THESE SUBROUTINES GET THE DATE & TIME FROM
                ;       A COMPU/TIME BOARD MANUFACTURED AT 8532 HAMILTON AVE.,
                ;       HUNTINGTON BEACH,CA.,92646,(714)536-9967
                ;
                ;       THE BOARD MUST BE ADDRESSED BEGINNING AT 'C0' OR YOU MUST
                ;       CHANGE THE PORT ASSIGNMENTS BELOW
                ;
                ;       THE ADDRESS OF THE DESTINATION OF THE DATE & TIME MUST BE IN
                ;       THE D&E REGISTER WHEN THESE ROUTINES ARE CALLED
                ;
                ;       A CALL TO TIME RETURNS HH:MM:SS  (8 ASCII BYTES)
                ;       A CALL TO DATE RETURNS MM/DD/YY  (    DITTO     )
                ;
                ;       THIS SOFTWARE EXECUTES ON THE CP/M OPERATING SYSTEM
                ;
                ;       WRITTEN BY W.C.HOFFER-2721 N. WANDA-SIMI VALLEY,CA.-93065
                ;
F000                    ORG     0F000H
                ;
                ;ASSIGNMENTS
                ;
00C5 =          ACONT:  EQU     0C5H    ;PORT A CONTROL
00C7 =          BCONT:  EQU     0C7H    ;PORT B CONTROL
00C4 =          ADATA:  EQU     0C4H    ;PORT A DATA
00C6 =          BDATA:  EQU     0C6H    ;PORT B DATA
F000 C341F0             JMP     DATE    ;GET THE DATE
F003 C364F0             JMP     TIME    ;GET THE TIME
                ;
                ;
                ;READ A DIGIT ROUTINE
                ;
F006 7A         RDIGIT: MOV     A,D     ;SELECT DIGIT
F007 D3C4               OUT     ADATA   ;
F009 DBC4               IN      ADATA   ;RESET INTERRUPT
F00B DBC5       DWAIT:  IN      ACONT   ;TEST FOR DIGIT PRESENT
F00D E680               ANI     80H     ;ANYTHING THERE?
F00F CA0BF0             JZ      DWAIT   ;LOOP UNTIL INTERRUPT
F012 DBC4               IN      ADATA   ;READ A DIGIT
F014 E60F               ANI     0FH     ;MASK ZONE
F016 F630               ORI     30H     ;SET ASCII
F018 C9                 RET
                ;
                ;READ FOUR DIGITS ROUTINE
                ;
F019 1600       READ4:  MVI     D,0     ;SET TO SELECT FIRST DIGIT
F01B CD06F0     NEXT:   CALL    RDIGIT  ;DELAY ONE DIGIT SCAN
F01E CD7FF0             CALL    RSDIG   ;READ & STORE DIGIT
F021 7A                 MOV     A,D
F022 FE20               CPI     20H     ;TEST IF 2 DIGITS DONE
F024 C237F0             JNZ     SKIP    ;
F027 79                 MOV     A,C     ;SEE IF TIME OR DATE
F028 FE00               CPI     0       ;DATE
F02A C232F0             JNZ     COLON   ;IT'S FOR TIME
F02D 3E2F               MVI     A,'/'   ;
F02F C334F0             JMP     DOIT    ;
F032 3E3A       COLON:  MVI     A,':'   ;
F034 CD3EF0     DOIT:   CALL    SDIGIT  ;
F037 7A         SKIP:   MOV     A,D     ;TEST FOR ALL DIGITS DONE
F038 FE40               CPI     40H     ;
F03A C8                 RZ              ;ALL 4 DONE
F03B C31BF0             JMP     NEXT    ;GET ANOTHER DIGIT
                ;
                ;STORE A DIGIT ROUTINE
                ;
F03E 77         SDIGIT: MOV     M,A     ;STORE A DIGIT
F03F 23                 INX     H       ;INCR H&L
F040 C9                 RET             ;
                ;
                ;READ DATE ROUTINE
                ;
                DATE:
F041 CDA5F0             CALL    BOARD   ;IS THERE A BOARD
F044 CA8AF0             JZ      NOBOARD ;NOPE
F047 EB                 XCHG            ;PUT ADDRESS OF DESTINATION IN H&L
F048 CDAAF0             CALL    CLKINT  ;INIT THE BOARD
```

```
FF4B 3E00          MVI   A,0      ;SET DATE DISPLAY MODE
FF4D D3C6          OUT   BDATA    ;
FF4F 0E00          MVI   C,0      ;TELL READ4 THIS IS DATE
FF51 CD19FF        CALL  READ4    ;GET 4 DIGITS
FF54 3E2F          MVI   A,'/'    ;
FF56 CD3EFF        CALL  SDIGIT   ;
FF59 3E37          MVI   A,'7'    ;SET TENS OF YEARS
FF5B CD3EFF        CALL  SDIGIT   ;
FF5E 3E38          MVI   A,'8'    ;SET UNITS OF YEARS
FF60 CD3EFF        CALL  SDIGIT   ;
FF63 C9            RET
            ;READ TIME & READ AND STORE DIGIT ROUTINES
            ;
FF64 CDA5FF TIME:  CALL  BOARD    ;CHECK FOR BOARD PRESENT
FF67 CA8AFF        JZ    NOBOARD  ;NOPE
FF6A EB            XCHG           ;PUT ADDRESS OF DESTINATION IN H&L
FF6B CDAAFF        CALL  CLKINT   ;INIT THE BOARD
FF6E 3E40          MVI   A,40H    ;SET TIME DISPLAY MODE
FF70 D3C6          OUT   BDATA
FF72 0E01          MVI   C,1      ;TELL READ4 THIS IS TIME
FF74 CD19FF        CALL  READ4    ;GET 4 DIGITS
FF77 3E3A          MVI   A,':'    ;
FF79 CD3EFF        CALL  SDIGIT   ;
FF7C CD7FFF        CALL  RSDIG    ;
FF7F CD0CFF RSDIG: CALL  RDIGIT   ;
FF82 CD3EFF        CALL  SDIGIT   ;
FF85 7A            MOV   A,D      ;
FF86 C610          ADI   10H      ;
FF88 57            MOV   D,A      ;
FF89 C9            RET            ;
            ;
            ;NO BOARD IN THE SYSTEM
            ;
            NOBOARD:
FF8A 3E30          MVI   A,'0'    ;STUFF 8 ZEROS
FF8C CD3EFF        CALL  SDIGIT   ;
FF8F CD3EFF        CALL  SDIGIT   ;
FF92 CD3EFF        CALL  SDIGIT   ;
FF95 CD3EFF        CALL  SDIGIT   ;
FF98 CD3EFF        CALL  SDIGIT   ;
FF9B CD3EFF        CALL  SDIGIT   ;
FF9E CD3EFF        CALL  SDIGIT   ;
FFA1 CD3EFF        CALL  SDIGIT   ;
FFA4 C9            RET            ;
            ;
            ;SEE IF BOARD PRESENT
            ;
            BOARD:
FFA5 DBC4          IN    ADATA    ;
FFA7 FEFF          CPI   0FFH     ;
FFA9 C9            RET            ;
FFAA 3E70  CLKINT: MVI   A,70H    ;
FFAC D3C4          OUT   ADATA    ;
FFAE 3E77          MVI   A,77H    ;
FFB0 D3C6          OUT   BDATA    ;
FFB2 3E14          MVI   A,14H    ;
FFB4 D3C5          OUT   ACONT    ;
FFB6 3E04          MVI   A,04H    ;
FFB8 D3C7          OUT   BCONT    ;
FFBA C9            RET            ;
```

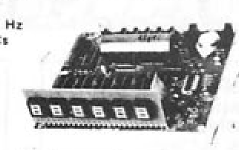
Many users prefer to employ a higher level language to communicate with their systems. Once the TIME-SUBS software has been added and the modifications have been made to BIOS, you can get the date and time from BASIC or FORTRAN or any other language you use, as long as you have the ability to interface with assembly language routines. I have used the date and time with both BASIC and FORTRAN, but since there are so many versions, I won't go into the details.

In conclusion, I believe that no system is complete without date and time. The uses are only limited by the need. I'm sure many of you have the need for date and time, and I urge you to try the software I've described. □

"Talk about a rough day. The teaching machine had a short circuit so I had to use something called a pencil and paper."

30 MHZ LOW COST FREQUENCY COUNTER KIT

Features:
- Frequency Range—100Hz to 30MHz min., resolution 100 Hz
- All TTL Circuitry—No tears in the eyes when replacing ICs
- FET Input Stage—Offers high input impedance
- High Sensitivity—15mV typical
- Xytal Time Base—0.001% 10MHz for better accuracy
- On Board Regulator—No external power supply needed
- All ICs Socketed—Easy to service
- Easy to Operate—No switches to flip
- Tin Plated & Screened Board—For easy assembly



COMPARE and SAVE!

$54.95

KIT # T-250-30A

KIT INCLUDES: Detailed Instructions (22 pages). All parts including transformer (case not available).

5V 10A Power Supply Kit for your TTL Circuits!

Kit Includes: Extra Large Heat Sink, Power Tr., IC Regulator, P.C. Board, with OVP Circuitry. KIT # T-500 **$14.95**

With Optional Rectifiers Filtering Capacitor. **$17.95**

Car Battery Eliminator

KIT INCLUDES: Transformer, PC Board, Large heat sink, Large filtering capacitor

KIT#700 **$16.95**

PRIME FROM ERIE

10-40p Mini Capacitance Trimmer

50 ea. 10/$4.50

2-20V ADJ. POWER SUPPLY

KIT INCLUDES: Transformer, PC Board, Large heat sink, Large filtering capacitor and all the parts with detailed instruction. KIT # T-458 **$12.95**

Instrument Case

for Kit H. 2 7/16" W. 8½ D. 9¼" and in 6 varying heights also compatible as other instrument housing

Mod. # CH200

Perfect for our 30 MHz Frequency Counter

PRICE: $17.50

6-DIGIT AUTO CLOCK KIT WITH ALARM

Features:
- A. Fairchild 0.5" FND 500 Series Display
- B. Display Board may be remote
- C. X'tal time base
- D. P.C. Boards, speaker, IC's and all parts.
- E. Detailed Instructions

$19.95 KIT # T-1302

4/6 Digit Alarm Clock Kit

Kit includes P.C. Boards, x'former and all parts. 4-Digit $10.95 6-Digit $12.95

I.C. SOCKETS

	Lo-pro 5 pcs	10 pcs	100 pcs	3-Level Wire Wrap 5 pcs	10 pcs	100 pcs
8-pin	0.80	1.50	12.00	1.90	3.50	32.00
14-pin	0.95	1.75	15.00	2.10	3.90	36.00
16-pin	1.05	2.00	17.00	2.50	4.20	39.00
18-pin	1.40	2.50	23.00	3.20	6.00	55.00
20-pin	1.60	3.10	28.00	3.50	6.50	60.00
22-pin	1.80	3.40	32.00	3.75	7.00	65.00
24-pin	1.85	3.50	33.00	4.25	8.00	75.00
28-pin	2.20	4.20	39.00	4.75	9.00	85.00
40-pin	3.50	6.20	59.00	6.50	12.00	100.00

HAMMOND BREADBOARD KIT

BIMBOARD I $9.95 ea.

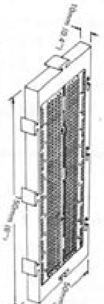
Accepts DIP packages without adaptors or damaging component leads.

Contacts are double sided, nickel silver, current carrying capacity of 1 Amp with less than 10 milliohms contact resistance.

Total of 550 sockets identified by a letter and number matrix for recording experiments

Buss strip section runs up each side of board.

Component bracket (included with each board) will fit on any of the four edges or down the center.



BIMBOARD 2

2 Bimboards and 2 component brackets

1 Aluminum base with 4 insulated Terminals

BIMBOARD 3

3 Bimboards and 3 component brackets

1 Aluminum base with 4 insulated terminals

BIMBOARD 4 $43.

4 Bimboards and 4 component brackets

1 Aluminum base with 4 insulated terminals

PRECUT WIRE WRAP WIRE

30 Kynar in red, orange, blue, yellow, green, black, white. Pre-stripped both end.

	100	500	1000	5000
2½ in.	.78	2.40	4.30/K	3.89/K
3 in.	.82	2.60	4.71/K	4.22/K
3½ in.	.86	2.80	5.12/K	4.55/K
4 in.	.90	3.00	5.52/K	4.88/K
4½ in.	.94	3.21	5.93/K	5.21/K
5 in.	.98	3.42	6.34/K	5.52/K
5½ in.	1.02	3.65	6.75/K	5.86/K
6 in.	1.06	3.85	7.16/K	6.19/K
6½ in.	1.15	4.05	7.57/K	6.52/K
7 in.	1.20	4.25	7.98/K	6.85/K
7½ in.	1.25	4.45	8.39/K	7.18/K
8 in.	1.29	4.65	8.80/K	7.53/K
8½ in.	1.32	4.85	9.21/K	7.84/K
9 in.	1.36	5.05	9.62/K	8.17/K
9½ in.	1.40	5.25	10.03/K	8.50/K
10 in.	1.45	5.51	10.44/K	8.83/K
Addl. in.	10	.41	.82/K	66/K

TRANSISTORS

- NPN-General Purpose 30V 10/$1.00
- PNP-General Purpose 30V 10/$1.00
- 2N2222-Switching 10/$1.50
- 2N3055-150W Power 10/$6.75
- 2N6059-Darlington Power, $3.25 ea. 20A HFE 1k Typical

Z-80 CPU BOARD/KIT

On board 2708 EPROM addressable to any 4K boundary above 32K. Power-on-jump to any 4K boundary above 32K, or the on board 2708. On board run-stop flip-flop and optional generation of Memory Write allow front panel-less operation.

BARE BOARD $34.00

$130.00 (2MHz) $145.00 (4MHz)

ASSEMBLED & TESTED ADD $50.00

8080 A CPU BOARD/KIT

With 8 level vector interrupt, CPU chip 8080 clock chip: 8224; crystal Freq.: 18MHz: vector interrupt chip: 8214.

BARE BOARD $28.50 KIT $95.00

Assembled and Tested $145.00

REAL TIME CLOCK FOR S-100 BUSS

On board 1MHz Crystal Oscillator; two independent interrupts. 16 bit counter in 10 s steps; and decade steps from 100 s to 10 sec. ASSEMBLED & TESTED $179.95

BARE BOARD $24.00

8K STATIC RAM

8K Kit 450ns $129.95 350ns $139.95 250ns $169.95

ASSEMBLED/TESTED ADD $25.00

16K STATIC RAM KIT $375

BOARD $24.00 ASSEMBLE/TEST $425.00

32K STATIC RAM

BARE BOARD $35.00

KIT		ASSEMBLE/TESTED
250ns	$795.00	$875.00
450ns	$625.00	$775.00

2708/2716 EPROM KIT $59.95

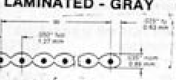
Special Kit With 8ea 2708's $129.95

Bare Board $30.00

ULTRAVIOLET EPROM ERASER

Erase up to 48 ICs each time $49.50

CABLE LAMINATED - GRAY 28 AWG 7/26 Strand



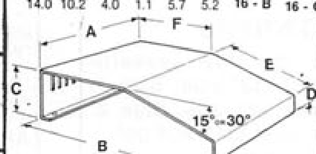
No. of Conductors	Width	Price/ft
10	0.45"	0.33
14	0.65"	0.46
16	0.75"	0.53
20	0.95"	0.66
26	1.25"	0.86
34	1.65"	1.15
40	1.95"	1.35
50	2.45"	1.65
60	2.95"	2.00

LAMINATED - COLOR CODED

No. of Conductors	Width	Price/ft
20	0.45"	0.54
14	0.65"	0.75
16	0.75"	0.86
20	0.95"	1.05
26	1.25"	1.40
34	1.65"	1.85
40	1.95"	2.15
50	2.45"	2.70
60	2.95"	3.25

MINIATURE CONSOLES

A	B	C	D	E	F	CASES (bottoms)				PANELS (tops)			
15° SLOPING PANELS													
4.0	5.5	2.0	1.1	3.3	2.2	1-B	1-G	1-L	5.50	1-W	1-S	1-K	5.50
6.5	5.5	2.0	1.1	3.3	2.2	2-B	2-G	2-L	6.10	2-W	2-S	2-K	6.10
6.5	8.5	2.0	1.1	3.3	5.2	3-B	3-G	3-L	6.70	3-W	3-S	3-K	6.70
6.5	8.3	3.0	1.3	6.3	2.2	4-B	4-G	4-L	7.30	4-W	4-S	4-K	7.30
10.0	8.3	3.0	1.3	6.3	2.2	5-B	5-G	5-L	7.80	5-W	5-S	5-K	7.80
10.0	11.3	3.0	1.3	6.3	5.2	6-B	6-G	6-L	8.40	6-W	6-S	6-K	8.40
14.0	8.3	3.0	1.3	6.3	2.2	7-B	7-G	7-L	8.80	7-W	7-S	7-K	8.80
14.0	11.3	3.0	1.3	6.3	5.2	8-B	8-G	8-L	9.70	8-W	8-S	8-K	9.70
30° SLOPING PANELS						B=Blue, G=Green, L=Gold, W=White, S=Sand, K=Black							
4.0	5.5	3.0	1.1	3.7	2.2	9-B	9-G	9-L	5.50	9-W	9-S	9-K	5.50
6.5	5.5	3.0	1.1	3.7	2.2	10-B	10-G	10-L	6.10	10-W	10-S	10-K	6.10
6.5	7.2	4.0	1.1	5.7	2.2	11-B	11-G	11-L	6.70	11-W	11-S	11-K	6.70
10.0	5.5	3.0	1.1	3.7	2.2	12-B	12-G	12-L	7.30	12-W	12-S	12-K	7.30
10.0	7.2	4.0	1.1	5.7	2.2	13-B	13-G	13-L	7.80	13-W	13-S	13-K	7.80
10.0	10.2	4.0	1.1	5.7	5.2	14-B	14-G	14-L	8.40	14-W	14-S	14-K	8.40
14.0	7.2	4.0	1.1	5.7	2.2	15-B	15-G	15-L	8.80	15-W	15-S	15-K	8.80
14.0	10.2	4.0	1.1	5.7	5.2	16-B	16-G	16-L	9.70	16-W	16-S	16-K	9.70



1N4148 Diodes

20 for 1.00
50 for 2.25
100 for 4.00

POWER TRANSISTORS

MATCHED PAIR

MOTOROLA MJE2955 PNP

MJE3055 NPN

10 AMP 60 VOLT 90 WATTS

$2.25 PER PAIR

MINIATURE SLIDE SWITCH

DPDT .20 each

10 for $1.75

100 for $15.00

MINIATURE TOGGLE SWITCH

SPDT $1.30 ea.

DPDT $1.50 ea.

PUSH BUTTON SWITCH

White, green and yellow 30¢ ea. 4/$1.00

0.2" L.E.D. Lamps

Red	15¢	100/10.00
Green	20¢	100/15.00
Yellow	20¢	100/15.00
Amber	20¢	100/15.00

L.E.D. Displays

END 503	C.C.	0.5"	75¢
END 510	C.A.	0.5"	75¢
1720R	C.A.	1"	4.75
1723R	C.C.	1"	4.75

JUMBO LED READOUT ARRAY

- 1.00 Inch High Characters
- Common Cathode Connections
- 3½ Digits with Colons
- Independent Segments

LT1012 $7.50

CRYSTALS ± =0.002%

Frequency MHz

1.000	2.4576	4.0000	6.144	10.000	32.0000
1.8432	2.667	4.1943	6.400	14.318	36.0000
@5.25	3.000	4.9152	6.5536	14.391	48.0000
2.000	3.200	5.0000	8.0000	18.000	@3.50
2.0100	3.2768	5.0688	@4.25	18.4320	
2.097152	3.5795	5.1850		20.0000	
	@4.75	5.7143		22.1184	
		6.0000		27.0000	

SOLDER W/K

Quick solder remover

1/32"	5 ft.	
1/16"	5	
1/8"	5	1.40 ea.
1/4"	2½	

TANTALUM CAP AXIAL LEADS

1	10V	15¢
1	35V	20¢
3.3	35V	25¢
10	50V	30¢
22	35V	30¢

KEPCO Ferroresonant Power Supply

Please add $5.00 extra postage to these items.

Input 110V 3A Factory Price $206

Output 24V 8A Our Price $80

LAMBDA

Precision Power Supply Model LXS-EE-5-OV

5V 45A ~~$600.~~ $100 ea.

INTER—COM BOARD.

Fully assembled. Works on 9-15V D.C. 2 speakers make it work. With Schematic

ONLY $3.00

DIP PLUGS

	5 pcs	10 pcs	100 pcs
14-pin	3.25	6.00	55.00
16-pin	3.50	6.50	60.00
40-pin	9.50	18.00	150.00

THE KIM-1 MICROCOMPUTER SYSTEM

- 6502 - 8 bit MPU with powerful instruction set, 13 addressing modes, multiple interrups, full 65K byte address range.
- 2 MCS 6530 chips each with 1024 bytes of ROM, 64K bytes of RAM, 15 I/O pins, and an interval timer.
- The KIM-1 monitor and operating programs are stored permanently in the 1048 ROM bytes provided.
- Comes with keyboard and alpha-numeric display.
- TTY and cassette interface.
- Completely assembled and tested and documented.
- Including KIM-1 user manuel, 6500 programming manual, and 6500 hardware manual.

ALL THESE FOR JUST $235.00

Wire Wrap Wire Special Purchase

500 ft. roll@ 5.00 ea. White only

KIM SUPPORT

6102	8.00
6502	11.95
6520	10.00
6522	9.25
6530	15.95
6530-002	15.95
6530-003	15.95
6530-004	15.95
6530-005	15.95

0.8" 4 Digit Jumbo Display Alarm Clock Kit

Features:
- A. Fairchild 0.8" FSC8000 Display Array
- B. Fairchild Super-Chip – F-3817PC
- C. P.C. Board, Transformer, Speaker and all parts included (less case)
- D. Detailed Instructions

$19.50

60 Hz Time Base IC

MM5369 $1.95

WIRE-WRAP TOOLS from OK

Hobby Wrap - 30 $5.45

Hobby Wrap—Model BW-630 Battery Op. (less batt.) $30.95

SPECIAL OF THE MONTH

AN214-4.5W Power IC with spec.	$2.95 ea.
MH0026-5MHz Clock Driver	$1.95 ea.
14-pin Lo pro IC Socket	100/$16.00
16-pin Lo pro IC Socket	100/$17.00
P.C. Mount X'former, 24V .45A	$1.95 ea.
1,000 uF 50V cap. Axial Lead	$1.00 ea
C-60 Digital Cassette Tape	$1.50 ea

TRINICO INTERNATIONAL

2795-L West Lincoln Ave.,
Anaheim, CA 92801.

Please add $1.00 postage inside Calif., $2.00 for out of State, Overseas orders, please add 10%. ☐ BankAmericard ☐ Master Charge

Min. Order $5.00 C.O.D. Orders $20.00

Store Hours: Mon - Sat. 10 - 7

3815 5-DECADE COUNTER

The 3815 is a 5-Decade Counter which includes a memory with static latches for each counter digit and an output multiplexer. The 3815 is designed to drive a multiplexed display which has a Binary Coded Decimal output and five decoded outputs to strobe the display.

With Specification $5.00 ea.

MICROCOMPUTER COMPONENTS

MICROPROCESSOR'S

F8	16.95
Z80	22.00
Z80A	28.00
CDP1802CD	19.95
2650	24.95
AM2901	22.95
6502	11.95
6800	18.95
6802	25.00
8008-1	9.95
8035	22.00
8080A	11.95
8085	27.00
TMS9900TL	75.00

8080A SUPPORT DEVICES

8212	3.50
8214	9.00
8216	3.75
8224	3.50
8224-4	9.95
8226	3.95
8228	7.95
8238	7.50
8251	9.95
8253	21.95
8255	21.95
8257	21.95
8259	21.95
8275	75.00
8279	20.00

6800 SUPPORT

6810P	4.95
68B10P	6.00
6820P	7.50
6821P	7.50
6828P	11.25
6834P	16.95
6850P	9.75
6852P	11.75
6860P	10.00
6862P	14.50
6871P	28.00
6875P	8.75
6880P	2.50

PROM'S

1702A	4.00
2704	15.00
2708	12.00
2716	30.00
2716 intl	38.00
2758	26.60
D3601	4.50
D3604	13.00
5203AQ	5.00
5204AQ	7.50
6834	17.50
6834-1	14.95
82S23B	4.00
82S129B	4.25
8223B	3.50

STATIC RAMS

2102	1.50
21L02	1.50
21L02 (350)	1.60
21L02 (250)	1.75
4100	10.75
1101A	1.00
2101-1	2.95
2102	1.25
2111-1	3.95
2112-1	2.95
2114-3	11.00
2125L	11.10
2147	37.50
3101	2.50
3106	3.95
3107	3.95
TMS-4044	9.95
4200A	12.95
TMS-4045	11.00
5001	8.30
74C89	3.25
7489	2.25
74S201	4.50
P8101	4.20
P8155	17.00
P8156	21.00
8599	1.88
91028PC	1.65

DYNAMIC RAMS

416D/4116	32.00
1103	1.00
2104	4.00
2107B	4.25
2107B-4	3.95
TMS4050	4.00
TMS4060	4.50
TMS4070-2	32.00
4096	4.00
4116/416D	32.00

Z80 SUPPORT DEVICES

3881	12.95
3882	12.95

USRT

S2350	10.75

USART'S

AY5-1013A	5.25
AY5-1014A	8.25
TR1602B	5.25
TMS6011	5.95
IM6402	10.80
IM6403	10.80

CHARACTER GENERATORS

2513	6.75
2513 5v upper	9.75
2513 5v lower	10.95
2525	10.95

KEYBOARD CHIPS

AY5-2376	13.75
AY5-3600	13.75

MICRO MARKET

D.C. METRO AREA
TYSONS CORNER, VIRGINIA
COMPUTER SYSTEMS STORE
Microcomputers & Peripherals
for Home, School, & Small Business
1984 Chain Bridge Rd., McLean, VA. 22101
703-821-8333

ISAM For MICROPOLIS

- MAGSAM allows you to Create, Retrieve, and Update data records by user defined keys
- Interactive Tutorial Program, File Dump Utility, and User Guide are also included
- Comprehensive User Guide includes generalized ISAM concepts ($5 ordered separately)
- Only $75 for complete MAGSAM package (Distributed on Mod II diskette)

MICRO APPLICATIONS GROUP
7300 CALDUS AVE, VAN NUYS, CA 91406

TRS-80 SOFTWARE 3/$10.00
1. Handicap Meadowland Trotters
2. Inventory 3. Tic Tac Toe
Medical Software: Available
P.O. Box 4092 Bergen Station
Jersey City, New Jersey 07304

HEATH
LISTINGS • TIME • TREK • DAY
CONVERT. NOS • BACKGAMMON
(401) 331-5034 — BASIC
(401) 738-3265 — LIBRARY

REAL TIME MONITOR

For Z80, 8080. Supervises multitasking environment with or without clock. Self-relocating, ROM-able, Paper tape, PT-cassette, CP/M disk. $35.00

F. A. STEARNS
2235 Frankfort St.
San Diego, CA 92110 (714) 222-6381

NEW SURPLUS PRINTER
Okidata – CP 110 – Friction Feed
Un-Used, in O.E.M. Factory Cartons.

- 5x7 Impact Dot Matrix
- 80 Char/Line
- 64 Char ASCII
- 110 Char/Sec.
- 66 Lines/Min.
- Accepts 8½" Roll paper

Includes – Power Supply, Built in Selftest, Parallel Interface, Line Buffer and Cables. Housed in a three piece plastic cabinet with all control electronics. Retail for over $1,100

Our Price: 1 ea. $650 10 up $600 ea.
Operating Manual Included.
Service Manual $20 *Supplies Limited*
Shipped Freight Collect. Send Check or M.O.
Sorry No Warranty Available at these Prices

INTERNATIONAL ELECTRONICS
EQUIPMENT CORP.
P.O. Box 522542, Miami, Florida 33152

COMPUTER HOBBYISTS!
Nationwide Classified
Advertising Newsletter
Buy & Sell Hardware & Software
new/used/unique/low-cost
$3.75 for 18 issues.
Free sample issue on request.
ON_LINE Newsletter
24695 Santa Cruz Hwy, Los Gatos, Ca. 95030

SHORTY CASSETTES

3M C-10 Low Noise High Denisty. Short enough to rewind and still hold many programs.
$1.50 ea. — 10 for $14.00 — Send $1.00 for complete supply catalog with refund coupon.

ELLIAM ASSOCIATES
24000 BESSEMER
WOODLAND HILLS, CA. 91367

UNEXPLAINED GLITCHES/CRASHES?

AC Power Line surges and hash could be the culprit! Our Surge Suppressors and Hash Filters curb damaging surges and disruptive hash. Send self-addressed, stamped envelope for FREE Surge and Hash Cures flyer. Get your Suppressor/Filter at your Dealer or order direct.

S/F-KW-3 1000 watt line cord unit	$20.95
ELF/T-10 10 amp wire-in unit	$17.25

ELECTRONIC SPECIALISTS, DEPT. IF
171 S. Main St., Natick, MA 01760

TRS-80 COMPUTING
non-profit newsletter
$10 (U.S.)/12 issues payable
Computer Information Exch., Inc.
Box 158
San Luis Rey, CA 92068

PET OWNERS

"PROGRAM," a monthly cassette "magazine" for PET, $27/year. Side 1: articles, peripherals, reviews. Side 2: games, programs. Just load and go!

PROGRAM
P.O. Box 461 • Philipsburg, PA 16866

NEW
Shugart Minifloppy SA400 ...... $285
Shugart Floppy SA800/1 .... $450
IBM Comp. 8" diskettes .... $3.50 ea.
Verbatim 5" diskettes .... $3.50 ea.
SAVE 10-50% ON NEW AND USED
COMPUTER EQUIPMENT
COMPUTER INTERFACE TECHNOLOGY
(714) 979-9920
Specializing in DEC, DG, Interdata

KIM-1 $219
With Power Supply $245
PET software and hardware
Write for list of KIM and PET accessories
A B Computers
P.O. Box 104, Perkasie, PA 18944

ATTENTION
NORTH STAR USERS

ARIAN: 8080 Operating System Monitor, Assembler, Editor, Debugger Save & Load Object/Source Files to Disc Resident and Custom Commands <><>$50.00<><>

INVESTMENT PACKAGE: Stocks, Bonds, etc. <><>$25.00<><>

FAMILY PACK: Programs for home and family <><>$25.00<><>

GAMES: Includes "Starfleet" Plus many others <><>$15.00<><>

SUPERSOFT
P.O. Box 1628, Champaign, IL 61820

Check our prices — We will not be undersold!

POLY-8813	1 drive-No monitor	$2500
POLY-88	System 2	$575
IMSAI I-8080	22 Slot	$575
IMSAI	PC Boards/Kits	15% Off
APPLE II	16K	15% Off
EXTENSYS	Memory	20% Off

AAAA Computer How's
1477 Barrington, Suite 17
W. Los Angeles, CA 90025 (213) 477-8478

6800 SOFTWARE

REAL TIME CLOCK — Software + instructions for adding a REAL TIME CLOCK to your computer. For under $3. in additional parts, this interrupt will give your SWTPC 8K BASIC, or machine language programs the time of day $7.50

GRAPH #1 — Create real-time graphics with ease from your machine language program — also generates ASCII characters! Uses the SWTPC GT-61 display $5.50

Each program requires a 6800 with MIKBUG, or compatible monitor. Memory required: under 2K. Full source listing and MIKBUG paper tape are included, or add $1.00 for Kansas City format cassette to each price above.

*We also have GAMES, UTILITY PROGRAMS, and a FORTRAN CROSS ASSEMBLER. Write for flyer!
*Coming soon, an enhanced Graphics package with ASCII, Russian, and Greek characters plus other features. Please write for details.

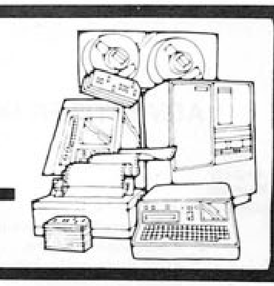
APPLIED MICROCOMPUTER SYSTEMS
Box 68, Silver Lake, NH 03875

2708 & 2716 E-PROM PROGRAMMER
WITH SOFTWARE
For KIM-1, Motorola 6800, Fairchild F-8, RCA 1802, or Intel 8080.
Assembled & tested, **$59.95. Kit $49.95**

Optimal Technology, Inc.
Blue Wood 127, Earlysville, VA 22936
After 1:00 P.M. (804) 973-5482

MICRO-MARKET AD RATE:
$50.00 per column inch. Maximum of 4 column inches per ad. Submit ads to:
Micro-Market Ads
INTERFACE AGE Magazine,
PO Box 1234
Cerritos, CA 90701.

FIFO FLEA MARKET



FOR SALE: North Star Disk Z80/8080 assembly language development system: relocating macro assembler, text editor, interactive assembler, trace debugger, linkage editor, relocating loader. Ready to run, full documentation. $140. Dealer inquiries invited. A. Ashley, 395 Sierra Madre Villa, Pasadena, CA 91107, (213) 793-5748.

WANTED: COSMAC VIP programs. Sell SWTP CT-1024 with extra optinos, $150. Joe Torzewski, 51625 Chestnut Rd., Granger, IN 46530.

WANTED: Correspondence with COSMAC 1802 users. I would like to share software and ideas. Karl Mosgofian, 1142 Treat St., San Francisco, CA 94110.

FOR SALE: Heath H-8 software: Payroll, billing, inventory, easy-to-use and adapt to your requirement, $20 each. Bingo, Blackjack, Bomber, Bowling, Checker, Craps, Derby, Dice, Football, Golf, Hangman, Hmrabi, Hi-Lo, Hockey, Lunar, Nim, Target, Tic-Tac-Toe, War and more, any 5 for $10. Investment, depreciation, loan and 20 other personal programs. All compatible to Heath extended BASIC. I will write any small programs to your spec. for $10. S. Kwong, 4361 Blackford Ave., 4, San Jose, CA 95129.

FOR SALE: Heath H-8 Computer System, H8-1 memory board (4K), H8-3 chip set (4K), H8-5 serial I/O & cassette interface (1 or 2 recorders), H8-2 parallel interface, ECP-3801 cassette recorder, H9 video terminal. Assembled and working, asking $1500. Thomas C. Smith, 506 19th Lane, Vero Beach, FL 32960, (306) 569-4622 after 6 P.M.

TRS-80 software available: Automobile Expense Record System, Complete Check Maintenance System, Complete game package available. If your request is not available, I'll write it. $15 each or write for more information. Will also design/write any program you need. 10 years experience. Robert L. Trenor, 4113 Ohio Ave., Tampa, FL 33616.

INTERESTED in exchanging commodities trend analysis programs with others who are both computer and commodity oriented. Write Bob Lemeer, P.O. Box 407, Little Neck, NY 11363.

FOR SALE: TRS-80 software on cassette: Tic-Tac-Toe, Slot Machine, Loan Payment, Day of the Week, Petals Around the Rose. All will run in 4K RAM, Level I BASIC, $6.00 each, 2 for $10.00 (ppd). Also have complete business system for Amway Distributors. Write for details. Circle Enterprises, 8 Driftwood Cir., Groton, CT 06340.

WANTED: for Microdata 1600 CPU (Reality): mag tape controller, disk controller and drive, core memory boards. Jack Hardman, 140 Forest Ave., Glen Ridge, NJ 07028, (201) 429-8880.

WANTED: To purchase: December 1975 and January 1976 INTERFACE AGE. Any offer considered. Felix Macri, Apt. 10, No. 8 Gloucester Rd., Hurstville, NSW 2220, Australia.

TRS-80 SORT: in BASIC I, 4K min. Sorts recs. in memory on 1 or 2 numeric fields in asc. or desc. seq. Input from tape, k.b., or both. Output to tape or video. Can be subroutine. $10 on tape w/doc. To: Microware, Box 6153, Syracuse, NY 13217.

FOR SALE: Professional "Power One" power supply 5v. 12a. current, foldback and overvolt. protected with data and schem. used 2 mos. $70 or best offer. L. Langrehr, Work: (312) 262-1600 Ext. 491, Home: 2069 N. Humboldt Blve., Chicago, IL 60647.

MUST SELL: 5-month old MITS Altair 8800A, includes 32K RAM, ACR, 2SIO, CPU, ADM-3A, manuals, Ext. BASIC on cassette, etc. System's excellent, no troubleshooting. $2,700 or best offer, plus free tape cassette recorder. First cashier's check or money order to M. Wedgle, 1090 S. Parker Rd., #A-216, Denver, CO 80231.

FOR SALE: Complete Altair disk system and controller. DOS with documentation and disk BASIC included. Minus only cable assembly. $780. Also, MITS TTY-Serial I/O board with documentation, $65. All have been assembled and operating. Make me an offer. M. Liebman, 3940 Ballina Cyn., Encino, CA 91436, Phone: (213) 986-5427.

TRS-80: American in Germany is lonely for other TRS-80 users. Swap programs, ideas, etc. How about a European/Overseas User's Group? We need hardware and software freaks. How about interfacing or robotics? Richard M. Bash, Sudeten Ring 69, 6072 Sreieich, Sprendlingen, Germany.

PET OWNERS: I have 13 game programs for the PET computer. I will trade 1 for 1 for your programs. Anyone wishing to trade should send their cassette to Kurt Huebner, 1266 Valle Vista, Fullerton, CA 92631.

WANTED: Game programs for TRS-80. Generous royalties. Please submit complete description. Your rights respected. Earl J. Bivins, Contract Services Associates, 1846 W. Broadway, Anaheim, CA 92804.

FOR SALE: Fully assembled and fully operational Morrow Micro Stuff/Thinker Toys front panel CPU board. All documentation included. First certified funds for $300 gets it including U.P.S. shipping charges. W. Howard Adams, 1590 So. Krameria St., Denver, CO 80224, (303) 756-4052.

WANTED: Person wanting to share in start-up and operation of new computer produce oriented company. Person will share in marketing, production, finance and management areas. Please respond in total confidentiality to E.C.W., P.O. Box 22664, Denver, CO 80222.

I AM LOOKING for an individual or company willing to provide on site service contracts for the metropolitan New York City area. Equipment would be microcomputers and related peripherals. Also looking for owners who would like to have on site service and maintenance for their equipment. B. Rabinowicz, 1061 54th St., Brooklyn, NY 11219.

BLACKJACK SIMULATOR for the TRS-80 in BASIC-I. Program allows you, with a minimum knowledge of BASIC, to write routines to test your 'system' for beating B.J. The simulator calls your routines at hit, ins. and bet time. Simulator plays thousands of hands and records statistics on tape. Package includes analysis program to display statistics. Progs. on tape w/doc. $20. Send to: Microware, Box 6153, Syracuse, NY 13217.

FOR SALE: ASR-33 excellent conditoin with stand, asking $350. Add $5.00 for schematic. Michael Solomon, 664 Belmont Ave., Montreal, Quebec, H3Y-2W2, Canada, (514) 484-1645 between 5 & 7.

SELECTRIC INTERFACE: Assembled controller (SC/MP + RAM and EPROM) and power supply to allow conversion of any office Selectric to an ASCII I/O terminal (RS232, 110 or 300 baud) with included solenoid and LED detector mechanisms. $450. J. R. Cameron, Dept. of Biochemistry, Stanford Medical Center, Stanford, CA 94305.

FOR SALE: MEK6800D2 system working with SWTPC TV Terminal III with 69 key Hall-effect keyboard. Includes 24 2102's, two extra 6810 RAMs, huge 6800 applications manual, data manuals, software manual, plus power supply kit. $295 or best offer. C. Jackson, 1715 Dogwood Dr., Rock Hill, SC 29730, (803) 366-2309.

SOUTHERN CONNECTIVUT: Anyone using Tarbell disk system please contact me for mutual system support. Also need help in maintenance, fine points of CP/M; will pay. James Van Pelt, 25 Sagamore Cove, Branford, CT 06405.

ADVERTISER INDEX

*Manufacturer requests factory-direct inquiry.

FOR SALE: SWTPC 6800 computer systems, 8K memory, serial control interface, parallel interface, CT-64 terminal, SD Sales cassette interface, Pixie-verter. All assembled, up and running, complete with documentation. Also, bare 4K memory board, bare MP-A2 processor board. Contact Rick Drapala, 2701 Avenue A, Yuma, AZ 85364, (602) 726-6896, after 4 P.M. MST.

FOR SALE: Two Lambda LT-2095M 0-32 volt, 0-2 amp power supplies $75 each. SWTP 143 0-35 volt, 0-2 amp power supply $20.00. Technison 10-inch (-15 to 0 to +15 volt input signal) chart recorders 115 volt 60 Hz power requirement, $200 each. S. Lei, P.O. Box 5312, Fargo, ND 58102.

WANTED: Sol/Processor Technology keyboard, with numerical pad. Roger Wallace, 36 Las Vegas Rd., Orinda, CA 94563, (415) 254-1248.

PROGRAMS FOR: Sol, Sol BASIC 8, North Star BASIC, Sol/North Star. Super D.O.S., personalized, D.O.S. for Sol. $5. Bomb, Suspense Game $4. Star-Ship Trainer, $4. Convert $2.50 Octal/Hex loader. $3. Plus many more. Above are listings. For cassette add $2.50, mini-disk $5.25. Send for complete list to Pete Pacione, 2952 N. Meade, Chicago, IL 60634.

FOR SALE: OSI 16K system, battery back-up on 4K, 430 I/O (A-D, two D-A, phase lock cassette), 445 video (4mhz xtal controlled), 495 proto (w/6 latched digits of 7 seg. readouts, sockets for PIAs etc), well regulated and filtered power supply, mother board, front panel, complete documentation and software included, $1290. Also avilable OSI 450 EPROM board with 1K EPROM (new and unbuilt), $70. R. W. Vogt, Box 240A, R.D.1, Port Matilda, PA 16870, (814) 237-8238.

Program Listing

Branched from page 82

```
>LIST
  1 REM  DISPLAY SELECTION
  2 DIM X(5),Y(5),B(10),A$(40),COL(6) X(1)=0:Y(1)=0
  5 X(2)=10:X(3)=20:X(4)=30:X(5)=39:Y(2)=12:Y(3)=24:Y(4)=36:Y(5
   )=47
  7 COL(1)=3:COL(2)=6:COL(3)=12:COL(4)=13:COL(5)=9:COL(6)=1
 10 CALL -936: TAB 8: VTAB 4: PRINT "T.V. PATTERN GENERATOR"  TAB
    14: VTAB 7
 20 PRINT "WRITTEN BY"  TAB 10: VTAB 10: PRINT "ROBERT E. HARR, JR."
     TAB 4: VTAB 20
 30 INPUT "(HIT RETURN KEY TO CONTINUE)  ",A$
 40 TEXT : CALL -936
 45 TAB 4: VTAB 4: PRINT "SELECT SCREEN DISPLAY BY NUMBER"  TAB
    4: VTAB 8
 50 PRINT "1..SOLID COLOR"  TAB 4: VTAB 10: PRINT "2..RAINBOW COLORS
    "  TAB 4: VTAB 12
 55 PRINT "3..DOT MATRIX"  TAB 4: VTAB 14: PRINT "4..VERTICAL LINES"
     TAB 4: VTAB 16
 60 PRINT "5..HORIZONTAL LINES"  TAB 4: VTAB 18: PRINT "6..CROSSHATC
    H"
 65 TAB 8: VTAB 6: INPUT " --->",B: IF B<1 OR B>6 THEN 80
 70 GR : POKE -16302,0: COLOR=0: FOR I=40 TO 47: HLIN E0,39 AT
    I: NEXT I
 75 COLOR=15:DISP=B*100: GOTO DISP
 80 VTAB 20: TAB 4: PRINT "BAD SELECTION, TRY AGAIN (1 THRU 6)"
     GOTO 65
100 REM  GENERATE SOLID COLOR
110 GR : N=3: FOR I=0 TO 15: COLOR=I : FOR M=1 TO 2: VLIN 0,39 AT
    N:N=N+1: NEXT M : NEXT I
115 TAB 1: VTAB 21: CALL -958: TAB 4
120 PRINT " 0 1 2 3 4 5 6 7 8 9101112131415"
125 TAB 4: VTAB 22
130 INPUT "SELECT COLOR (0 THRU 15)",C
135 IF C>=0 AND C<16 THEN 150: TAB 4: VTAB 23
140 INPUT "BAD SELECTION, TRY AGAIN!",C: GOTO 135
150 COLOR=C: POKE -16302,0: FOR I=0 TO 39: VLIN E0,47 AT I : NEXT
    I
160 INPUT A$: GOTO 40
200 REM  RAINBOW COLORS
205 N=1
210 FOR L=1 TO 6: COLOR=COL(L)
220 FOR I=1 TO 6:M=N+I : VLIN E0,47 AT M
230 NEXT I: N=N+6: NEXT L
240 INPUT A$ : GOTO 40
300 REM  DOT MATRIX GENERATOR
310 FOR I=1 TO 5: FOR J=1 TO 5: PLOT X(J),Y(I)
320 NEXT J: NEXT I: INPUT A$: GOTO 40
400 REM  VERTICAL LINE GENERATOR
410 FOR I=1 TO 5: VLIN E0,47 AT X(I): NEXT I
420 INPUT A$: GOTO 40
500 REM  HORIZONTAL LINE GENERATOR
510 FOR I=1 TO 5: HLIN E0,39 AT Y(I): NEXT I
520 INPUT A$ : GOTO 40
600 REM  CROSSHATCH GENERATOR
610 FOR I=1 TO 5: VLIN E0,47 AT X(I)
620 HLIN E0,39 AT Y(I) : NEXT I
630 INPUT A$ : GOTO 40
```

READER SERVICE CARD 1

INTERFACE AGE™

August 1978 Issue
Void after Nov. 30, 1978
(Please type or print)

Name____________________ Title__________

Company ____________ Phone (A/C) __________

Address ____________________ ☐Home ☐Business

City ____________ State_____ Country_____ Zip______

ANSWER THE FOLLOWING BY CHECKING ONLY ONE BOX PER QUESTION.

1. I Am A:
A. ☐Professional (Medical, Accounting, Law, Etc.)
B. ☐Engineer (Electronics, Mechanical, Etc.)
C. ☐Business Person (Retail, Wholesale, Etc.)
D. ☐Educator (Professor, Teacher, Assistant, Etc.)
E. ☐Student
F. ☐Hobbyist
G. ☐Other________

2. My status is:
A. ☐Have all computing equipment
B. ☐Need more peripherals
C. ☐Have CPU only
D. ☐Have no equipment
E. ☐Other________

3. I need this information for:
A. ☐Immediate purchase
B. ☐Purchase 30-60 days
C. ☐Comparisons
D. ☐Literature Library
E. ☐Other________

4. My interest emphasis is:
A. ☐Hardware
B. ☐Firmware
C. ☐Software
D. ☐Other________

5. My application is:
A. ☐Hobby only
B. ☐Business only
C. ☐Hobby & Business only
D. ☐Instruction Purposes
E. ☐Research
F. ☐Design & Development
G. ☐Other________

6. My primary source of "State-of-the-Art" information comes from:
A. ☐Magazines
B. ☐Exhibits & Conventions
C. ☐Club meetings
D. ☐Direct mail from manufacturers
E. ☐Other________

7. I prefer to buy:
A. ☐Directly from manufacturers
B. ☐Local computer retailer (store)
C. ☐Mail order
D. ☐Club group purchase
E. ☐Other________

8. I look to INTERFACE AGE first for:
A. ☐New product information
B. ☐Software information
C. ☐Tutorials
D. ☐Hardware articles
E. ☐Product advertising
F. ☐Remarks________

Please send information on items circled below:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75
76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125
126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150
151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200
201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224	225
226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243	244	245	246	247	248	249	250
251	252	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275
276	277	278	279	280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300
301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320	321	322	323	324	325
326	327	328	329	330	331	332	333	334	335	336	337	338	339	340	341	342	343	344	345	346	347	348	349	350
351	352	353	354	355	356	357	358	359	360	361	362	363	364	365	366	367	368	369	370	371	372	373	374	375
376	377	378	379	380	381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400

READER SERVICE CARD 2

INTERFACE AGE™

August 1978 Issue
Void after Nov. 30, 1978
(Please type or print)

Name____________________ Title__________

Company ____________ Phone (A/C) __________

Address ____________________ ☐Home ☐Business

City ____________ State_____ Country_____ Zip______

ANSWER THE FOLLOWING BY CHECKING ONLY ONE BOX PER QUESTION.

1. I Am A:
A. ☐Professional (Medical, Accounting, Law, Etc.)
B. ☐Engineer (Electronics, Mechanical, Etc.)
C. ☐Business Person (Retail, Wholesale, Etc.)
D. ☐Educator (Professor, Teacher, Assistant, Etc.)
E. ☐Student
F. ☐Hobbyist
G. ☐Other________

2. My status is:
A. ☐Have all computing equipment
B. ☐Need more peripherals
C. ☐Have CPU only
D. ☐Have no equipment
E. ☐Other________

3. I need this information for:
A. ☐Immediate purchase
B. ☐Purchase 30-60 days
C. ☐Comparisons
D. ☐Literature Library
E. ☐Other________

4. My interest emphasis is:
A. ☐Hardware
B. ☐Firmware
C. ☐Software
D. ☐Other________

5. My application is:
A. ☐Hobby only
B. ☐Business only
C. ☐Hobby & Business only
D. ☐Instruction Purposes
E. ☐Research
F. ☐Design & Development
G. ☐Other________

6. My primary source of "State-of-the-Art" information comes from:
A. ☐Magazines
B. ☐Exhibits & Conventions
C. ☐Club meetings
D. ☐Direct mail from manufacturers
E. ☐Other________

7. I prefer to buy:
A. ☐Directly from manufacturers
B. ☐Local computer retailer (store)
C. ☐Mail order
D. ☐Club group purchase
E. ☐Other________

8. I look to INTERFACE AGE first for:
A. ☐New product information
B. ☐Software information
C. ☐Tutorials
D. ☐Hardware articles
E. ☐Product advertising
F. ☐Remarks________

Please send information on items circled below:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75
76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125
126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150
151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200
201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224	225
226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243	244	245	246	247	248	249	250
251	252	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275
276	277	278	279	280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300
301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320	321	322	323	324	325
326	327	328	329	330	331	332	333	334	335	336	337	338	339	340	341	342	343	344	345	346	347	348	349	350
351	352	353	354	355	356	357	358	359	360	361	362	363	364	365	366	367	368	369	370	371	372	373	374	375
376	377	378	379	380	381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400

First Class
Permit No. 217
Clinton, Iowa

BUSINESS REPLY MAIL

No Postage Stamp Necessary if Mailed in the United States

Postage will be paid by

INTERFACE AGE™ Magazine

Box 2654

Clinton, Iowa 52732

First Class
Permit No. 217
Clinton, Iowa

BUSINESS REPLY MAIL

No Postage Stamp Necessary if Mailed in the United States

Postage will be paid by

INTERFACE AGE™ Magazine

Box 2654

Clinton, Iowa 52732

ATTENTION DEALERS:

Jim-pak electronic components — One-Stop Component Center

NEW!! EXPANDED PRODUCT LINE . . .

TTL

7400	7490
7402	7492
7404	7493
7408	74107
7410	74121
7420	74122
7430	74123
7432	74125 (DM8093)
7442	74154
7447	74161 (DM9316)
7473	74176 (DM8280)
7474	74177 (DM8281)
7475	74192
7476	74193
7485	74367 (DM8097)

Low Power TTL Schottky

74LS00	74LS73	74LS109
74LS02	74LS74	74LS123
74LS04	74LS75	74LS136
74LS08	74LS83	74LS138
74LS10	74LS85	74LS151
74LS20	74LS86	74LS175
74LS30	74LS90	74LS367
74LS32		

C/MOS

4000	4020	4050
4001	4023	4051
4010	4024	4069
4011	4029	4071
4013	4044	4081
4016	4046	4511
4017	4049	

Display LEDS Discrete

MAN2	XC556-Red
DL704	XC556-Green
DL707	XC556-Yellow
DL747	CLIPLITES—Red, Green, Yellow
DL750	

LINEAR

LM301V	LM567V
LM307V	LM723N
LM309K	LM739N
LM311V	LM741V
LM320K-5 (7905)	LM747N
LM320T-5 (7905)	LM1458V (5558V)
LM320T-12(7912)	LM1488N
LM320T-15(7915)	LM1489N
LM339N	LM1556V
LM340T-5 (7805)	XR2206
LM340T-12(7812)	CA3080
LM340T-15(7815)	CA3130
LM555V	CA3140
LM556N	LM3900N (CA3401)

MICROPROCESSOR

Z-80	2101
8080A	2102
8212	21L02
8224	7489
8228	MM5262
6800	1702A
6810	82S23
6830L8	2708
AY-5-1013	DM8835N
2513/2140	N8T97
MM5314(Clock Chip)	

SOCKETS

8 pin low profile	14 pin wire wrap
14 pin low profile	16 pin wire wrap
16 pin low profile	24 pin wire wrap
24 pin low profile	40 pin wire wrap
40 pin low profile	TO-3 Socket
14 pin plug	TO-5 Socket
16 pin plug	Molex Pins

SWITCHES

Dipswitches	Slide
Toggle	Push Button
Subminiature Push Button	

CAPACITORS

Aluminum Electrolytic

1mfd 50V	100mfd 50V
4.7mfd 50V	220mfd 50V
10mfd 50V	470mfd 50V
22mfd 50V	1000mfd 25V
47mfd 50V	2200mfd 16V

Ceramic Disc

10pf 50V	.001mfd 50V
47pf 50V	.0047mfd 50V
100pf 50V	.01mfd 50V
220pf 50V	.022mfd 50V
330pf 50V	.047mfd 50V
470pf 50V	.1mfd 50V

Dipped Tantalum

.1mfd 35V	2.2mfd 25V
.22mfd 35V	3.3mfd 25V
.33mfd 35V	4.7mfd 25V
.47mfd 35V	6.8mfd 25V
.68mfd 35V	10mfd 25V
1mfd 35V	15mfd 25V
1.5mfd 35V	33mfd 25V

Polyester Mylar

.001mfd 100V	.022mfd 100V
.0015mfd 100V	.047mfd 100V
.0022mfd 100V	.1mfd 100V
.0047mfd 100V	.22mfd 100V
.01mfd 100V	

CRYSTALS

CY1A CY2A CY12A

TEST CLIPS

14 pin clip 16 pin clip

FUSE HOLDERS

HKP-3AG

P.C. POTENTIOMETERS

Single-Turn	15-Turn
840P1K	830P1K
840P5K	830P5K
840P10K	830P10K
840P50K	830P50K
840P100K	830P100K
840P1meg	830P1meg

DIODES

IN751	IN4148(IN914)
IN4733	IN4001
IN4734	IN4004
IN4742	IN4007
IN4744	MDA-980-3

TRANSISTORS

C106B1	2N3055
2N2222A	2N3904
2N2907A	2N3906
MJE2955	2N5129
MJE3055	2N5139

CONNECTORS

DB25P Plug DB25S Socket

DATA BOOKS*

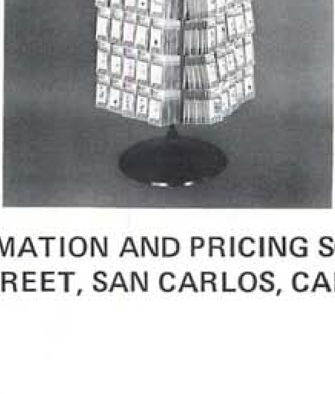
7400/74LS Data Book
CMOS/Linear Data Book
Microprocessor/LED Data Book
*JIM-PAK Products only

HEAT SINKS

¼ W RESISTOR ASST.

¼ Watt 5%
50 Pc. Assortments

- ⋆ Over 260 quality items including integrated circuits, resistors, diodes, transistors, capacitors, connectors, switches, sockets, crystals, test clips, fuseholders, potentiometers, heat sinks, LEDs and Data Books covering all JIM-PAK® items.
- ⋆ Immediate delivery on all orders
- ⋆ Store display racks available
- ⋆ Stock rotation and return policy
- ⋆ Direct mail program available from list of active electronic buyers in dealers' area.
- ⋆ National advertising campaign in leading electronics magazines to include list of qualifying dealers
- ⋆ Nationally known manufacturers' products at prices every dealer can afford
- ⋆ Guaranteed products
- ⋆ Standard industry part numbers



- A component line of proven sellers developed for the independent dealer. Ideal for computer shops, school stores, electronic dealers, hobby shops or any location where there is a potential market for electronic sales.
- A product line which supplies most of your needs from one distributor with a reputation for fast and efficient service. Attractive and compact display racks make initial installation of the JIM-PAK® line easy.
- Your customers deserve the best. Now you can profitably retail name brand components at competitive prices. Be the first in your area to announce and sell the JIM-PAK® line. Write or call today.

FOR MORE INFORMATION AND PRICING SCHEDULE CONTACT:
jim-pak electronic components — 1021 HOWARD STREET, SAN CARLOS, CALIFORNIA 94070 (415) 592-8097

CIRCLE INQUIRY NO. 27

INTRODUCING THE G/2 PROGRAM LIBRARY.

Now you don't have to be a programmer to use your computer!

At last there's a reliable, professional source of useful and entertaining programs for your personal computer. The G/2 Program Library from GRT unlocks the full power of your personal computer. Without any programming knowledge on your part.

The G/2 Program Library consists of applications programs pre-recorded on cassette tapes. A detailed manual with each cassette provides complete instructions. And G/2 programs contain prompting instructions. So they're easy to use every step of the way. Just load the tape and G/2 puts your computer to work.

The first three G/2 program tapes are available now: For exciting family games, get **Beat the House.** You'll find Blackjack, Craps, Roulette and Slot Machine programs, with truly random chances and realistic, casino-like payoffs.

Interested in your health? You'll want **Clinic.** Biorhythms, Dieting and Longevity programs give you new perspectives in health and happiness, along with some pleasant guidelines to follow.

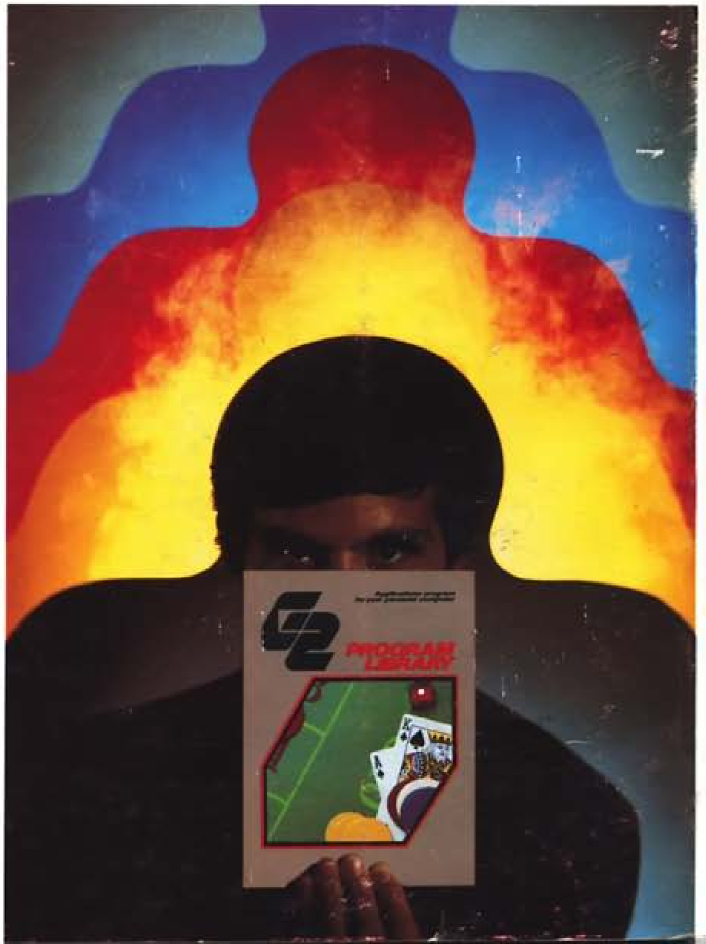
Need a better handle on your money? Buy **Dollars & Sense.** Checkbook is a comprehensive program to balance your account and keep within your budget, while Best Choice is a novel way to make decisions based on the facts you know and the relative importance of each. Use it for everything from choosing a new car to selecting your next stereo.

Soon, every dealer that handles personal computers will stock the G/2 Program Library. For the G/2 dealer nearest you, call us toll-free at 800/538-1770. (In California, 800/662-9810.) And if your favorite computer dealer doesn't yet offer G/2, have him call the same number.

Soon you'll see G/2 program tapes in many other fields of interest. The G/2 Program Library is now available for use with SOL, SWTPC, Exidy, TRS-80 and Apple personal computers. Just select the package marked for the computer you own.

Start building your G/2 Program Library now. And put the full power of your computer to work for you.

THE REASON YOU BOUGHT YOUR COMPUTER. G/2™



GRT® CORPORATION

A product of GRT Corporation
Consumer Computer Group
1286 North Lawrence Station Road
Sunnyvale, California 94086
(408) 734-2910

CIRCLE INQUIRY NO. 24